U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards

NBSIR 86-3385-4

Implementation Agreements for Open Systems Interconnection Protocols

NBS Workshop
for Implementors of
Open Systems Interconnection

REVISED MARCH 1987

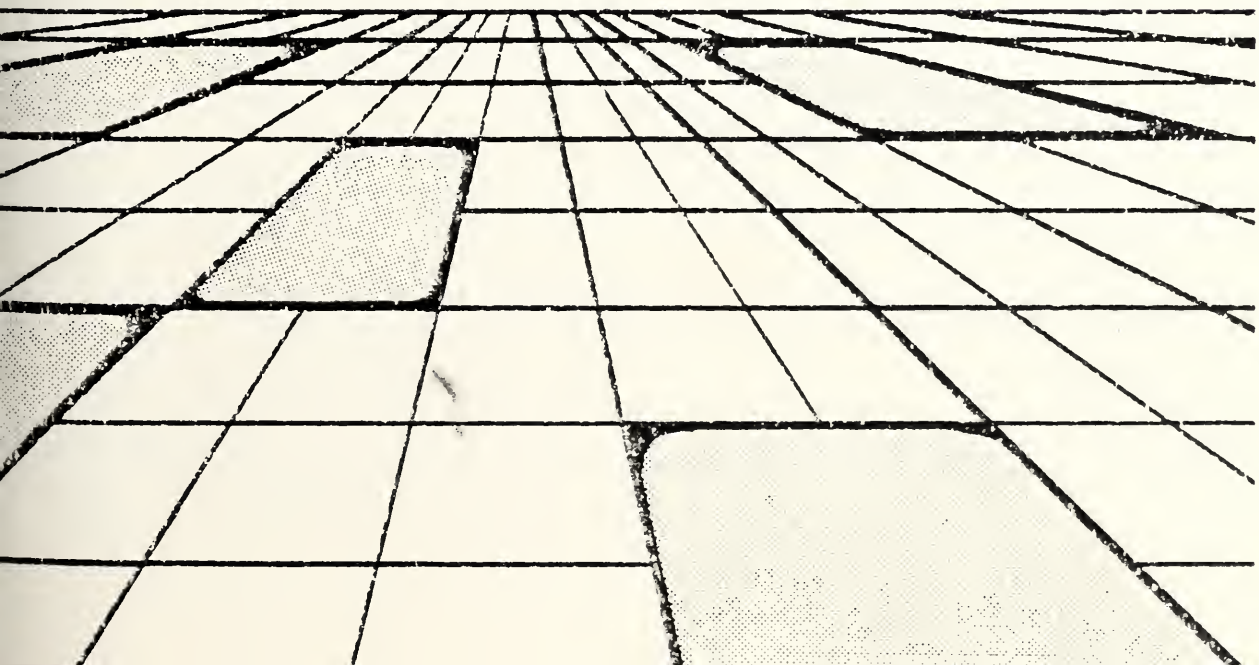

Table of Contents

1. GENERAL INFORMATION

1.1 PURPOSE OF THIS DOCUMENT

This document records ongoing implementation specification agreements of OSI protocols among the organizations participating in the NBS/OSI Workshop Series for Implementors of OSI Protocols. This work is not considered advanced enough for use in product development or procurement reference.

The companion document, "Final Implementation Agreements for Open Systems Interconnection Protocols," records completed agreements. As each protocol specification is completed, it will be moved from this document to the companion document.

1.2 PURPOSE OF THE WORKSHOP

In February, 1983 NBS organized the above named workshop to bring together future users and potential suppliers of OSI protocols. The workshop accepts as input the specifications of emerging standards for protocols and produces as output agreements on the implementation and testing particulars of these protocols. This process is expected to expedite the development of OSI protocols and promote interoperability of independently manufactured data communications equipment.

1.3 USE AND ENDORSEMENT BY OTHER ENTERPRISES

The workshops are held for those organizations expressing an interest in implementing or procuring OSI protocols and open systems. However, there is no corporate commitment to implementations associated with workshop participation.

The agreements contained in earlier versions of this document were used for OSI demonstrations at the National Computer Conference in 1984 and at the AUTOFACT conference in 1985.

The agreements from several versions of this document have been adopted for use in implementations running on OSINET.

The MAP/TOP Steering Committee has endorsed these agreements and will "continue the use of the most current, applicable Implementors Workshop Agreements in all releases of the MAP and TOP specifications."

The COS Strategy Forum has "adopted a resolution stating that as a matter of policy COS should select as its sources of Implementation Agreements organizations or forums that are: (1) Broadly open, widely recognized OSI workshops (NBS/OSI Workshops are first preference) ..."

The U.S. Government OSI User's Committee is using these implementation specifications in its Federal procurement specification, "The primary source of protocol specifications used in GOSIP is the Implementation Agreements for

Open Systems Interconnection Protocols. ... By primary source, it is meant that where GOSIP uses a given protocol, it cites that protocol by reference as specified in the above named Workshop Agreements."

1.4 RELATIONSHIP OF THE WORKSHOP TO THE NBS LABORATORIES

As resources permit, NBS, with voluntary assistance from industry, develops formal protocol specifications, reference implementations, tests and test systems for the protocols agreed to in the workshops. This is work made available to the industry volunteers and to others making valid commitments to organized events and activities such as NCC, AUTOFACT, and OSINET. As soon as this work can be adequately documented it is placed in the public domain through submission to the National Technical Information Service. Any organization may then obtain the work at nominal charge.

The NBS laboratories bear no other relationship to the workshop.

1.5 STRUCTURE AND OPERATION OF THE WORKSHOP

1.5.1 Plenary

The main body of the workshop is a plenary assembly. Any organization may participate. Representation is international. NBS prefers for the business of workshops to be conducted informally, since there are no corresponding formal commitments within the workshop by participants to implement the decisions reached. The guidelines we follow are: 1) one vote per company or independent division, 2) only companies that regularly attend should vote, 3) only companies that plan to sell or buy a protocol should vote on its implementation decisions, 4) only companies knowledgeable of the issues should vote, and 5) no proxy votes are admissible.

1.5.2 Special Interest Groups

Within the workshop there are Special Interest Groups (SIGs). The SIGs receive their instructions for their technical program of work from the plenary. The SIGs meet independently, usually during the workshop. As technical work is completed by a SIG, it is presented to the plenary for disposition. Companies participating in a SIG are expected to participate in the plenary. Voting rules for SIGS are the same as voting rules for the plenary.

Special Interest Groups sometimes correspond with organizations performing related work, such as ANSI committees. Such correspondence should be sent through the plenary to the parent committee, such as ANSC X3T5 or ANSC X3S3. When SIG meetings take place between workshops, the correspondence from these meetings should be addressed directly to the parent committee and copied to the workshop plenary.

Following are procedures for cooperative work among Special Interest Groups.

- Any SIG (SIG 1) or individual having issues to discuss with or requirements of another SIG (SIG 2) should bring the matter to the attention of the chairperson of that SIG (SIG 2).

- o The SIG 2 chairperson should bring the matter before SIG 2 for action.

- o SIG 2 should respond to the concerns or needs of SIG 1 or the individual in a timely manner.

- o If the matter cannot be satisfactorily resolved or if the request is outside the charter assigned to SIG 1, then it should be brought before the plenary.

- o SIGs are expected to complete work in a timely manner and bring the results before the plenary for disposition. However, the plenary may elect to act on any issue within the scope of the workshop at any time.

Following are the charters of the nine Special Interest Groups.

FTAM SIG

Develop Phase 2 product-level specifications.

Future new work items will be defined in a Phase 3 specification. It will contain only extensions of Phase 2 FTAM. It is a goal that Phase 3 will be backward compatible with Phase 2 FTAM. The set of future work items listed below may be changed by the plenary if the work is more appropriate for other SIGs.

High priority work items:

- o Clean up section 10 of this document

- o Specify Reliable File Service

- o Specify Recovery and Restart Data Transfer functional units in the user correctable file service

- o Specify concurrency control parameter.

Low priority work items:

- o Add new document types/constraint sets

- o Define subset of authorization requirements

- o Specify Presentation Context Management functional unit.

X.400 SIG

Develop product-level specifications for Message Handling Systems using the CCITT X.400 Recommendations.

Develop abstract tests for X.400, as requested by the ad hoc rapporteur for

this study question in CCITT. This work is to be submitted by the plenary (after its approval) to the U.S. Department of State as a proposed U.S. contribution to CCITT Study Group VII.

Lower Layer SIG

The Lower Layer SIG will study OSI layers 1-4 and produce recommendations for implementations to support the projects undertaken by the workshop and the work of the other SIGs. Both connectionless and connection-oriented modes of operation will be studied. The SIG will accept direction from the plenary for work undertaken and the priority which it is assigned.

The objectives of the Lower Layer SIG are:

- o Study OSI layers 1-4 as directed by the plenary,
- o Produce and maintain recommendations for implementation of these layers,
- o Where necessary, provide input to the relevant standards bodies concerning layers 1-4, in the proper manner, and
- o Begin work on the implementation specification of the ISO Network Layer Routing Exchange Protocol prior to the ISO draft achieving DIS status.

Performance SIG

The plenary will provide the following inputs to the OSI Performance SIG:

- o the set of applications for which the performance of OSI protocols is of particular concern,
- o the requirements for each application including:
  - performance targets
  - network topology
  - background network loads
  - application traffic characteristics, and
- o the final and ongoing, "Implementation Agreements Among Implementors of OSI Protocols".

The objectives of the OSI Performance SIG are to:

- o determine whether the OSI protocols are able to meet these performance requirements,

- o report these determinations to the plenary, and where appropriate, provide input to the voluntary standards bodies concerning changes to existing standards and the requirements for new ones, in the appropriate form.

OSI Security Architecture SIG

GOAL: To develop an overall OSI Security Architecture which is consistent with the OSI reference model and which economically satisfies the primary security needs of both the commercial and Government sectors.

APPROACH: To define a security architecture encompassing the security addenda presently being specified at certain OSI layers, the required cryptographic algorithms and related key management functions, and the security management functions which must be performed between the layers and the peer entities defined in the OSI architecture.

Directory Services SIG

Produce functional implementation agreements based on ISO/CCITT specifications for Directory Services in accordance with the objectives and goals of the plenary.

- o Provide a subset for NBS publication which is functional and forward compatible to further work by this Special Interest Group.

- o Define stable core functionality which can be implemented in the near term.

Virtual Terminal SIG

This Special Interest Group's charter is based upon the implementation of Draft International Standards 9040 and 9041 and their respective addenda, in providing Basic Virtual Terminal Service.

This group will develop agreements for the implementation and testing of the following terminal types.

- o X.29 PAD
- o TELNET
- o Basic Scrolling
- o Basic Paging
- o Basic Forms

Upper Layers SIG

The charter of the Upper Layers SIG is as follows.

- Develop product level specifications for the implementation of:
  - Session service and protocol,
  - Presentation service and protocol,
  - ACSE service and protocol.

- In addition, the specifications to be developed by the Upper Layers SIG will address issues that are common to layers 5-7 such as addressing, registration, etc. This SIG will review output and proposals from other SIGs to ensure consistency with international standards regarding Upper Layer Architecture.

- The specifications developed will be done to support the requirements of FTAM, X.400, VT, Directory Services and any other SIG.

The objectives of the Upper Layers SIG are to:

- Study OSI layers Session, Presentation, and ACSE,

- Incorporate implementor's agreements in the 1987 NBS standing document,

- Produce and maintain recommendations for implementations of these layers,

- Where necessary provide input to the relevant standards bodies concerning layers Session, Presentation, and ACSE,

- React in a timely manner (i.e., to develop corresponding implementor's agreements) to technical changes in ISO documents.

The following are the guidelines under which the Upper Layers SIG will operate:

- Align implementation agreements with other organizations such as ANSI and ISO,

- Develop implementor's agreements that promote the efficiency of protocols,

- Develop implementor's agreements that promote ease in the verification of interoperability,

- Develop necessary conformance statements.

Office Document Architecture and Office Document Interchange Format SIG

Develop product-level specifications for the architecture and interchange of

office documents processed by computers. The standard governing the ODA/ODIF SIG's work is ISO 8613 - Office Document Architecture (ODA) and Interchange Format (ODIF).

1.6 POINTS OF CONTACT

OSI Workshop - Chairman	Rob Rosenthal, NBS, 301/975-3603
OSI Workshop - Registration	Sara Arneson, NBS, 301/975-2934
FTAM SIG	Klaus Truoel, GMD/DFN, 49-615-1 869312
X.400 SIG	John Stidd, Xerox, 408/737-4338
Lower Layers SIG	Mike Gerring, IBM, 919/543-0481
Performance SIG	Mary Jane Strohl, CDS,617/460-0808
Security SIG	Denny Branstad, NBS, 301/975-2913
DS SIG	J. J. Cinecoe, WANG, 617/967-5514
VT SIG	Henry Lowe, DEC, 617/493-2572
Upper Layers SIG	Mike Ellis, HP, 916/786-8000x4292
ODA/ODIF SIG	Frank Dawson, McDonnell Douglas, 314/232-5251
MAP	Gary Workman, GM, 313/947-0599
TOP	Laurie Bride, BCS, 206/763-5719
Government OSI Profile	Jerry Mulvenna, NBS, 301/975-3631
OSINET	
Steering Committee	Jerry Mulvenna, NBS, 301/975-3631
Technical Committee	Ed Strum, IBM, 415/855-7392
SME (MAP/TOP Sponsorship)	Mark Shaw, 313/271-1500
	Paul Borawski, 313/271-1500
U.S. Government OSI User's Committee	Jerry Mulvenna, NBS, 301/975-3631

1.7 EVOLVING STRUCTURE OF THE DOCUMENT

This document is currently undergoing restructuring in order to:

- Allow new work to be incorporated as it is prepared by the SIGs, while setting apart and identifying those parts that are completed,
- Simplify the writing of procurement specifications referencing appropriate sections of the document and aiding the development of implementations of the protocols described herein, and
- Encourage completeness of each specification through consistency of format.

The plan is to make each protocol version a separate section. For example, the FTAM specifications based upon the draft proposed standard, the Draft International Standard, and the International Standard will become three separate sections. The structure for each section that is being worked toward is shown below.

Introduction

Scope and Field of Application

Status

Errata

Services

Protocol

Conformance

Appendices

The Introduction should introduce the protocol and reference the international documents from which the implementation specification is derived. The Scope and Field of Application should say when and how it is to be used or not used. The Status should say when the implementation specification is expected to be completed, or when it was completed. Once completed, the implementation may not be enhanced. Errors, when found, are to be corrected in the Errata section with each correction dated. Services describe the services provided and those assumed from the underlying layers. The Protocol section gives the implementation specifications for the protocol. At the time of the restructuring of the document, the services and protocols are somewhat intertwined, and thus may appear as one or a number of sections preceding the conformance section. The conformance section states what must be implemented and under what conditions, to be in conformance with the specification.

The remainder of the document is not completely in this form now, and parts may never be. When convenient old parts will be converted to this format and new protocol implementation specifications will be formatted in this way.

3. LOCAL AREA NETWORKS

3.1 IEEE 802.2 LOGICAL LINK CONTROL

The following decisions have been reached with respect to this protocol.

1. Link Service Access Point (LSAP)

The IEEE 802 committee has assigned the code below to address systems using ISO IS 8473 connectionless network protocols. Note that bit zero is transmitted first.

The most significant bit is bit 7, thus this bit pattern represents hexadecimal FE.

0	1	2	3	4	5	6	7
0	1	1	1	1	1	1	1

Fig. 3.1 LSAP bit pattern

2. Type and Class

Only the connectionless type 1, class 1 IEEE 802 link service will be used.

3.2 IEEE 802.3 CSMA/CD ACCESS METHOD

The 48 bit addressing will be used with the 10 megabit/second baseband coaxial cable specification.

3.3 IEEE 802.4 TOKEN BUS ACCESS METHOD

The following options are agreed to with respect to Draft F of token bus. An asterisk means that the option has been approved. The absence of an asterisk means that the option has not been approved.

1. Repeaters
   - Active Regenerative
2. Medium
   - Single Cable Coax *
   - Dual Cable Coax
3. Trunk Cable
   - RG-6 *
   - RG-11 *
   - Semi-rigid *
   - Other 75 ohm cables *
4. Trunk Connection Unit
   - 75 ohm tee connector
   - 75 ohm nondirectional passive

impedance-matching tap
75 ohm directional passive impedance-matching tap *

5. Transmit Carrier Frequency
   RF *
   Baseband
6. Modulation
   Phase Continuous FSK
   Phase Coherent FSK
   AM/PSK *
7. Encoding
   Manchester
   Duobinary *
8. Data Rate
   1 Mb
   5 Mb *
   10 Mb *
9. Addressing
   2 octet
   6 octet *
10. Connector at Station
    50 ohm Male BNC Series
    75 ohm Female F Series *
11. Priority (4 levels) *
12. Group Addressing *
13. Station Management
14. Broadband Channel Assignments

Forward	Reverse	
P	3'	*
Q	4'	*
R	4M'	*
S	5'	*
T	6'	*
U	FM1'	*

3.4 IEEE 802.5 Token Ring Access Method

The following implementation agreements are under consideration by the Lower Layers SIG, but formal approval has not yet been given.

- The data signalling rate shall be 4 MBPS
- The address length shall be 48 bits
- The message priority (PM) of the AMP data unit shall be 7
- The ALL_STATIONS_THIS_RING_ADDRESS shall be X'COOOFFFFFFFF'
- The TRR value shall be 4 milliseconds
- The THT value shall be 8.9 milliseconds

- The TQP value shall be 20 milliseconds
- The TVX value shall be 10 milliseconds
- The TNT value shall be 2.6 milliseconds
- The TAM value shall be 7 seconds
- The TSM value shall be 15 seconds
- Maximum frame -- the maximum MAC information field shall be 4425 bytes
- Minimum frame support requirement -- all stations shall support a MAC information field of at least TBD bytes.

4. WIDE AREA NETWORKS

4.1 CCITT RECOMMENDATION X.25

When providing CONS, it is agreed to use X.25 as the standard wide area network protocol. Elements of X.25 are explained in section 6.

5. PRIVATE SUBNETWORKS

5.1 PRIVATE SUBNETWORKS

The architectures agreed upon allow the use of private subnetworks in addition to private X.25 subnetworks. No particular private subnetwork has been discussed.

6. NETWORK LAYER

6.1 INTRODUCTION

This chapter presents agreements for providing the OSI network service. Also contained here are agreements on network layer addressing and routing.

6.2 SCOPE AND FIELD OF APPLICATION

These agreements cover both connectionless-mode and connection-mode network services.

6.3 STATUS

Completed in March 1987.

6.4 ERRATA

6.5 CONNECTIONLESS-MODE NETWORK SERVICE (CLNS)

6.5.1 Provision of CLNS Using ISO 8473

ISO 8473, Protocol for Providing the Connectionless-mode Network Service, will be used to provide the connectionless-mode network service. The full conformance protocol will be used with the following exceptions.

- The inactive subset for intra-subnetwork communication will not be supported. Implementations will not transmit PDUs encoded using the inactive subset. Received PDUs encoded using the inactive subset will be discarded.

- The non-segmenting subset will not be used. Implementations will not generate PDUs without a segmentation part. However, implementations will receive and correctly process PDUs which do not contain the segmentation part.

6.5.2 Agreements on Mandatory Protocol Functions

- An end system must provide a local means to control the value to be assigned to the lifetime parameter for PDUs which it originates.

6.5.3 Agreements on Optional Protocol Functions

- The Security parameter will not be used. Implementations should not transmit the Security parameter. If a received PDU contains the Security parameter, the PDU will be discarded. An ER PDU will not be generated.

- Source Routing will not be supported.[1]

- Record of Route will be supported by Intermediate systems.

- ISO 8473 will be followed with respect to QOS.

6.5.4 Subnetwork Dependent Convergence Function

A subnetwork dependent convergence function (SNDCF) for operating the CLNS over CCITT Recommendation X.25 has been agreed to. It shall adhere to the following.

- Conform to ISO 8473 AD1.

- The default throughput class should be used if this facility is available.

6.6 CONNECTION-MODE NETWORK SERVICE (CONS)

There is interest among a limited set of participants in implementing the connection-mode network service. This section records the agreement of the workshop on the provision of such a service.

When providing the CONS, the following shall apply.

- The definition of the CONS is as specified in ISO 8348, Network Service Definition.

- The mapping of the elements of the CONS to the elements of the X.25 Packet Level Protocol (PLP) is as specified in ISO 8878, Use of X.25 to Provide the Connection-mode Network Service.

- The general procedures and formats of the X.25 PLP are as specified in ISO 8208, X.25 Packet Level Protocol for Data Terminal Equipment.

6.6.1 Provision of CONS Using X.25/PLP-1984

The following agreements have been reached concerning the use of ISO 8878.

- The Receipt Confirmation service will not be provided, so the corresponding protocol elements need not be implemented.

- The Expedited Data service will not be provided, so the corresponding protocol elements need not be implemented.

---

[1]A problem exists with the Partial Source Routing option which can cause PDUs to loop in the network until their lifetime expires.

- Where the ISO 8208 diagnostic codes are not provided, all Cause/Diagnostic code combinations can be mapped to the Originator/Reason code of "Undefined".

6.6.2 Subnetwork Dependent Convergence Protocol

A subnetwork dependent convergence protocol (SNDCP) shall be used to provide the CONS in cases where an End system may not use the elements of the X.25/PLP-1984 needed to do so. This may be the case, for example, when operating in a packet-switched network environment which will treat as an error the use of any of the CCITT-specified DTE facilities.

The SNDCP to be used is defined in Annex A of ISO 8878 and referred to as the Alternative Procedures for Network Connection Establishment and Release.

The following agreements have been reached concerning the use of the SNDCP.

- The Receipt Confirmation service will not be provided, so the corresponding protocol elements need not be implemented.

- The Expedited Data service will not be provided, so the corresponding protocol elements need not be implemented.

6.6.3 User of X.25 for Connection-Oriented MHS

The following agreements are applicable only to MHS.

- Implementations operating in an X.25 1984 environment shall support OSI CONS as specified in 6.6.1.

- Implementations operating in an X.25 1980 environment may support OSI CONS as specified in 6.6.2.

- Implementations operating in a Non-OSICONS environment may use X.25 1980 only.

6.7 ADDRESSING

Address formats supported will conform to ISO 8348 DAD2 .

- NSAP address formats will have a hierarchical structure. This will reduce the size of routing tables.

- If used in the Domain Specific Part (DSP), an NSAP selector must be the least significant component in the hierarchy and does not affect routing. The NSAP selector is simply used to identify the network service user at the destination End system.

- There is a single NSAP selector for each NSAP within an End system. All NSAP addresses identifying a given NSAP will use the same NSAP selector value.

6.8 ROUTING

6.8.1 Static Routing

End systems and Intermediate systems supporting static routing will provide a local mechanism to update and, if necessary, to create the local routing table. Updating and consistency checking will be performed by human operators. The algorithms and data structures used for static routing are not specified in these agreements. Implementors are free to perform these functions in the manner which is most appropriate to their system environment.

6.8.2 End System to Intermediate System

Dynamic routing between End systems and Intermediate systems is performed using the protocol described in ISO 9542, End System to Intermediate System Routing Exchange Protocol for use in conjunction with ISO 8473. The following agreements apply to the use of this protocol over LANs and point-to-point links.

1. Implementations must support any valid NSAP format. For the purposes of the protocol, NSAP addresses are treated simply as octet strings.

2. Implementations must support both Configuration Information and Route Redirection Information. No subsets are permitted.

3. All timer values must be settable using local system management.

4. Use of checksums must be settable using local system management. Under normal use, checksums will be disabled.

5. The QOS, Security and Priority parameters should not be used for routing. For conformance, Intermediate systems must transmit these parameters in RD PDUs if they are present in the data PDU which generated the redirect. However, End systems must ignore them in received RD PDUs.

6. Both ES and IS implementations must support the 'optimization' described in Clause A.3 of ISO 9542 for system initialization. Its use must be selectable using local system management.

7. This protocol employs the same LSAP as ISO 8473.

8. The encoding of the BSNPA address follows the syntax rules for the data link being used. On a LAN, for example, it is a 48-bit MAC address.

6.9 CONFORMANCE

6.10 TEST REQUIREMENTS

THIS IS A BLANK PAGE.

THIS IS A BLANK PAGE

7. TRANSPORT

7.1 INTRODUCTION

These agreements support the integration of LANs, packet networks, and other WANs with the smallest possible set of mandatory protocol set's, in accordance with the other agreements already reached. Nothing here shall preclude vendors from implementing protocol suites in addition to the ones described in this document.

7.2 SCOPE AND FIELD OF APPLICATION

Two connection oriented transport classes have been identified for implementation (class 0 and class 4). In addition, there is interest among a limited set of participants in implementing a connectionless transport protocol. Transport class 4 (over CLNP) has been endorsed for general communication between private systems. Transport class 0 (over X.25) is used for communication with public (i.e., PT&T and RPOA) MHS systems operating in accordance with the CCITT X.400 series Recommendations. Communicating entities between private MHS systems over an X.25 network can, by negotiation or bilateral agreement, agree to use transport class 0. The connectionless transport protocol can be used with transaction-type implementations.

7.3 STATUS

Completed March 1987.

7.4 ERRATA

7.5 TRANSPORT CLASS 4

7.5.1 Transport Class

The following agreement has been reached with respect to this protocol.

Class 4 will be used with the required implementation of the 31 bit sequence space and 16 bit window size. The full protocol will be used including expedited data and negotiation at connection establishment.

7.5.2 Protocol Interpretation

According to the ISO transport specification, a disconnect request is issued in response to a connect request when the maximum number of transport connections is reached or exceeded.

7.5.3 Rules for Negotiation

- In general, the ISO rules for negotiation will be used, specifics follow.

- All implementations will send the 16/31 window size/sequence space in the CR TPDU. Implementations must all provide the 16/31 ISO option. Implementations must be able to accept the 4/7 in a CR TPDU.

- The ISO maximum TPDU size is negotiable between 128 and 8K octets, always negotiated downward. The ISO rules are to be followed, allowing any valid size in the CR TPDU. TPDU size negotiation is a local implementation issue. Each vendor will decide how it is implemented in their end system.

- The security parameter is optional and user defined in the ISO specification. Implementations should not send the security parameter in the CR TPDU; if received the parameter should be ignored.

- Both transports must agree to not use checksum, according to the ISO specifications. Requesting its use is an implementation choice. All implementations must be able to operate with checksum if requested.

- Use of acknowledgement time parameter is optional in ISO 8073. If an implementation is operating any policy which delays the transmission of AK TPDUs, the maximum amount of time by which a single AK TPDU may be delayed shall be indicated to the peer transport service provider using the acknowledgement time parameter. The value transmitted should be expressed in units of milliseconds and rounded up to the nearest whole millisecond.

- Throughput, priority, and transit delay are optional in the ISO specification. Do not send in the CR TPDU; ignore in the CC TPDU.

- User data in the CR TPDU and the CC TPDU are optional. No implementation should send; all implementations must be prepared to receive.

- An unknown parameter in any received CR TPDU shall be ignored.

- Known parameters with invalid values in a CR TPDU shall be handled as follows:

Parameter	Action
TSAP id	Send DR TPDU
TPDU size	ignore parm, use default
Version	ignore parm, use default
Protection (Security)	implementation dependent
Checksum	discard CR TPDU
Additional Options	Protocol Error
Alternate Protocol Classes	Protocol Error
Acknowledge Time	ignore parm

Throughput	ignore parm
Residual Error Rate	ignore parm
Priority	ignore parm
Transit Delay	ignore parm

7.5.4 Retransmission Timer

It is recommended that the value used for the retransmission timer be based upon the round-trip delay experienced on a transport connection. The implementation should maintain, and continually update, an estimate of the round-trip delay for the TC. From this estimate, a value for the retransmission timer is calculated each time it is started. An example technique for maintaining the estimate and calculating the retransmission timer is described below. Further information on similar techniques may be found in the literature [Edge 84, Jain 85, Mill 83].

The value of the retransmission timer may be calculated according to the following formula:

$$t \longleftarrow kE + w$$

In this formula, E is the current estimate of the round-trip delay on the transport connection, w is the value of the acknowledgement time parameter received from the remote transport service provider during connection establishment, and k is some locally administered factor.

A value for k should be chosen to keep the retransmission timer sufficiently small such that lost TPDUs will be detected quickly, but not so small that false alarms are generated causing unnecessary retransmission.

The value of E may be calculated using an exponentially weighted average based upon regular sampling of the interval between transmitting a TPDU and receiving the corresponding acknowledgement. Samples are taken by recording the time of day when a TPDU requiring acknowledgement is transmitted and calculating the difference between this and the time of day when the corresponding acknowledgement is received. New samples are incorporated with the existing average according to the following formula.

$$E \longleftarrow E + (1 - \alpha)S$$

In this formula, S is the new sample and alpha is a parameter which can be set to some value between 0 and 1. The value chosen for alpha determines the relative weighting placed upon the current estimate and the new sample. A large value of alpha weights the old estimate more heavily causing it to respond only slowly to variations in the round-trip delay.

A small value weights the new sample more heavily causing a quick response to variations. (Note that setting alpha to 1 will effectively disable the algorithm and result in a constant value for E, being that of the initial seed.)

If alpha is set to $1-2^{-n}$ for some value of n, the update can be reduced to a subtract and shift as shown below.

$$E \longleftarrow E + 2^{-n} (S - E)$$

When sampling, if an AK TPDU is received which acknowledges multiple DT TPDUs, only a single sample should be taken being the round-trip delay experienced by the most recently transmitted DT TPDU. This attempts to minimize in the sample any delay caused by the remote transport service provider withholding AK TPDUs.

7.5.5 Keep-Alive Function

The Class 4 protocol detects a failed transport connection by use of an 'inactivity timer'. This timer is reset each time a TPDU is received on a connection. If the timer ever expires, the connection is terminated.

The Class 4 protocol maintains an idle connection by periodically transmitting an AK TPDU upon expiration of the 'window timer'. Thus, in a simple implementation, the interval of one transport entity's window timer must be less than that of its peer's inactivity timer, and vice versa. The following agreements permit communicating transport entities to maintain an idle connection without shared information about timer values.

- In accordance with ISO 8073, clause 12.2.3.9.a, all implementations must respond to the receipt of a duplicate AK TPDU by transmitting an AK TPDU containing the 'flow control confirmation' parameter.

- Implementations must always transmit duplicate AK TPDUs on expiration of the local window timer (see ISO 8073, clause 12.2.3.8.1). Receipt of this TPDU by the remote transport entity will cause it to respond with an AK TPDU containing the 'flow control confirmation' parameter. When this is received by the local transport entity, it will reset its inactivity timer. See figure 7.1.

- It is a local matter for an implementation to set the intervals of its timers to appropriate relative values. Specifically:

    - The window timer must be greater than the round-trip delay. See section 7.1.4.

    - The inactivity timer must be greater than two times the window timer; and should normally be an even greater multiple if the transport connection is to be resilient to the loss of an AK TPDU.

A duplicate AK TPDU (See Figure 7.1.) is one which contains the same values for YR-TU-NR, credit, and subsequence number as the previous AK TPDU transmitted. A duplicate AK TPDU does not acknowledge any new data, nor does it change the credit window.

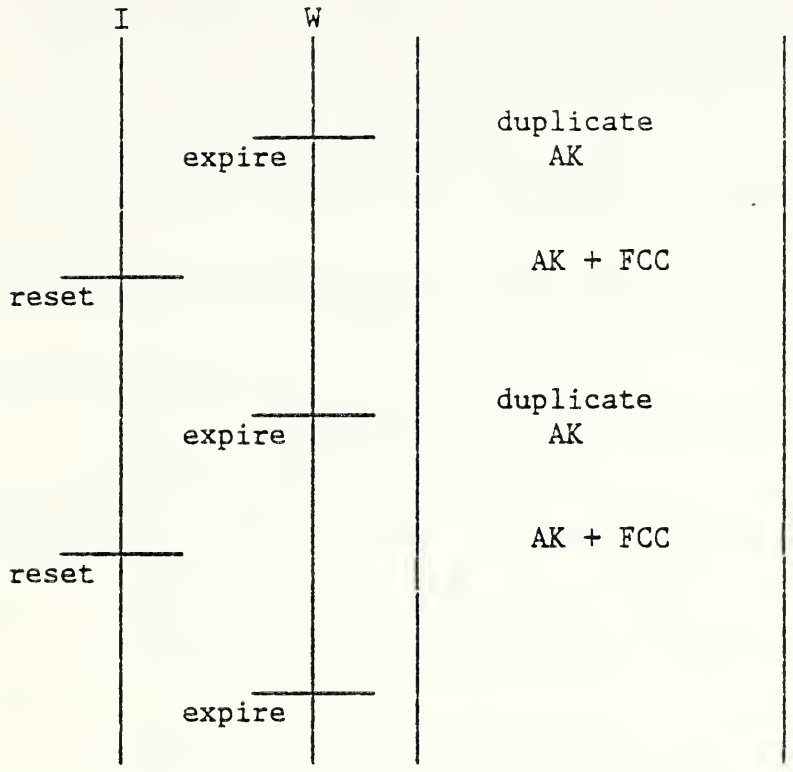

Fig. 7.1 AK exchange on idle connection

7.6 TRANSPORT CLASS 0

7.6.1 Transport Class

Transport class 0 over X.25 is mandatory (see X.400) for use in communicating with public MHS systems operating in accordance with the CCITT X.400 series recommendations. The purpose of the agreements concerning transport class 0 is to allow connection to these public services. Transport class 0 over X.25 can also be used in communicating between PRMDs (this choice is prevalent outside North America), but it is agreed that transport class 4 and CLNP over all types of lower layer networks allows a larger range of interoperation and is recommended but not required for systems conforming to these agreements.

7.6.2 Protocol Interpretation

Transport class 0 is a relatively simple protocol providing little opportunity for conflicting interpretations. A few relevant agreements follow.

- The Disconnect Request (DR) TPDU shall be limited to the first seven octets - "LI" plus "fixed part".

- The Error (ER) TPDU may be used at any time and upon receipt requires that the recipient disconnect the network connection, and by extension the transport connection.

- The allowed values for the maximum TPDU size are as specified in ISO 8073. They are: 128, 256, 512, 1024, and 2048.

- The class 0 protocol does not support multiplexing. At any instant, one transport corresponds to one network connection.

- It is recommended that the optional timers TS1 and TS2, if implemented, be settable by local system management. Values in the order of minutes should be supported.

- An unlimited TSDU length must be supported.

7.6.3 Rules for Negotiation

The ISO rules for negotiations will be used.

7.7 CONNECTIONLESS TRANSPORT

Document ISO IS 8072/DAD1 is the Transport Service Definition covering Connectionless-mode Transmission. Document ISO DIS 8602 is the Protocol for providing the Connectionless-mode Transport service.

8. UPPER LAYERS

8.1. INTRODUCTION

In this portion of the Implementors' Agreements, the NBS Upper Layers SIG is primarily concerned with providing implementation agreements for ACSE, and the Presentation and Session layers, so that systems implemented according to these agreements can successfully interoperate.

8.1.1. References

The following documents are referenced in the statement of the agreements relating to the NBS Upper Layers Architecture.

[1] Information processing systems-Open Systems Interconnection-Service definition for common application-service- elements-Part 2 Association control - ISO/DIS 8649/2 - 1986-04-29.

[2] Information processing systems-Open Systems Interconnection-Protocol specification for common application-service-elements- Part 2: Association control - ISO/DIS 8650/2 - 1986-04-29.

[3] Information processing systems-Open Systems Interconnection-Specification of Abstract Syntax Notation One (ASN.1) - ISO/DIS 8824 - 1986-09-10.

[4] Information processing systems-Open Systems Interconnection-Specification of Basic Encoding Rules for Abstract Syntax Notation (ASN.1) - ISO/DIS 8825 - 1986-09-10.

[5] ISO International Standard - 7498: "Information Processing Systems - Interconnection - Basic Reference Model" ISO/TC97/SC21. DP on Addendum on Naming and Addressing. Output from editing meeting <23-25 June 1986>.

[6] Application Layer Structure, ISO/97/21/N1494.

[7] ISO Presentation Draft International Standard: "Information Processing Systems - Open Systems Interconnection - Connection Oriented Presentation Service Definition." ISO/DIS 8822 - 1986-06-12.

[8] ISO Presentation Draft International Standard: "Information Processing Systems - Open Systems Interconnection - Connection Oriented Presentation Protocol Definition." ISO/DIS 8823 - 1986-06-12.

[9] ISO SESSION International Standard: "Information Processing Systems Interconnection - Basic Connection Oriented Session Service Definition." ISO/TC97/SC21 8326. Date: September 1984.

[10] ISO SESSION International Standard: "Information Processing Systems Open Systems Interconnection - Basic Connection Oriented Session Protocol Definition." ISO/TC97/SC21 8327 Date: September 1984.

[11] Information Processing Systems - OSI - Transport Protocol Specification, ISO/IS 8073, 1984.

[12] Protocol Rules for Extensibility, X3T5.5 85-543

[13] ISO International Standard - 7498: "Information Processing Systems - Interconnection - Basic Reference Model" ISO/TC97/SC21. First Edition - Oct. 15, 1984. Ref. No. ISO 7498-1984(E).

8.2. SCOPE AND FIELD OF APPLICATION

This section does not detail particular conformance statements for ACSE, Presentation, and Session, since what is to be implemented in each case depends on which Application Service Elements (ASE's) and which functional units within each ASE are used with an Application Process. Each ASE's SIG must specify which functional units of each layer it requires. However, the scope of each layer is based on the total indicated requirements of all application ASE's for which there is an active NBS SIG. The implementation agreements are not specified beyond that scope.

It is not the intent of this document to specify or reproduce standards, but, when a referenced standard is unclear or has known defects, an attempt to remedy the problem will be made herein. Any attempted clarification should be considered a possible interpretation; the ISO standard still takes precedence if there is any conflict. The situation with respect to defects in a standard is somewhat different; a reported defect may be technically resolved by the appropriate international technical committee, and approval by the voting members may be quite likely, but the vote may not occur for several months. Since relevant defects can't be ignored, this document will recommend using defect resolutions which have the tentative approval of the appropriate standards committees.

8.3. STATUS

This document is ready to be submitted to the NBS membership for review and a vote on its inclusion in the "NBS Implementation Agreements for OSI protocols" document.

8.4. ERRATA

8.4.1. ISO Defect Reports

This section lists the defect reports from ISO which have not yet been accepted but which are currently recognized to be valid for the purposes of NBS conformance.

8.4.1.1. Session Defects

There is a known defect in the Session Protocol State tables to the collision of FINISH SPDUs: For technical resolution refer to ISO/TC 97/SC 21/N486 Rev. Annex A revised.

8.5. ASSOCIATION CONTROL SERVICE ELEMENT

8.5.1. Introduction

This section details the implementation requirements for the Association Control Service Element (ACSE) of the Application layer. It is the intent of this section to follow the ISO ACSE standards. Where those specifications are inadequate, this section should provide the necessary information.

8.5.2. Services

8.5.2.1. ACSE Services

The following ACSE service primitives are within the possible scope of an NBS conformant system:

1. A_ASSOCIATE request
2. A_ASSOCIATE indication
3. A_ASSOCIATE response
4. A_ASSOCIATE confirm
5. A_RELEASE request
6. A_RELEASE indication
7. A_RELEASE response
8. A_RELEASE confirm
9. A_ABORT request
10. A_ABORT indication
11. A_P_ABORT indication

8.5.2.2. Use of Presentation Layer Services

ACSE services will make use of Presentation layer services as defined in reference [2].

8.5.3. Protocol agreements

Implementations will be based on references [1] and [2].

8.5.3.1. Application Contexts

Specific Application Contexts and their names will be supplied and defined by the appropriate NBS SIG.

Other application contexts may be defined and specified as dictated by particular application requirements.

8.6. PRESENTATION

8.6.1. Introduction

This section details the implementation requirements for the Presentation layer. It is the intent of this section to follow the ISO Presentation Standards. Where those specifications are inadequate, this section should provide the necessary information. The task of the Presentation layer is to carry out the negotiation of transfer syntaxes and to provide for the transformation to and from transfer syntaxes. The transformation to and from a particular transfer syntax is a local implementation issue and is not discussed within this section. This section is concerned with the protocol agreements, and thus is entirely devoted to the issues involved with the negotiation of transfer syntaxes and the responsibilities of the Presentation protocol.

8.6.2. Services

8.6.2.1. Presentation Services

Presentation Functional Units

The following functional units are within the possible scope of an NBS conformant system:

Presentation Kernel - This functional unit supports the basic Presentation services required to establish a Presentation connection, transfer normal data, and release a Presentation connection. This is a non-negotiable functional unit.

The Context Management and Context Restoration functional units are not within the scope of an NBS conformant system and need not be supported.

Any service not supported by the Session layer is also not supported by the Presentation layer; see the section on Session Functional Units for the possible Session functional units. The services provided by the Presentation layer are limited by the services provided by the Session layer as defined in [9] and [10].

8.6.2.2. Use of Session Layer Services

Presentation layer services shall make use of Session layer services as defined in reference [8].

8.6.3. Protocol Agreements

Implementations shall be based on references [7] and [8].

P-selectors are encoded as a string of octets, the meaning of which is known only to the local system. The length of the string cannot exceed 16 octets.

8.6.3.1. Transfer Syntaxes

The following transfer syntax must be supported for all mandatory abstract syntaxes: NBS-TS1.

This syntax is derived by applying the ASN.1 encoding rules to the abstract syntax (see reference [4]).

8.6.3.2. Abstract Syntaxes

Several abstract syntax names may map onto a single transfer syntax name. Note: the specific abstract syntax names are outside the scope of this Presentation specification and must be determined by the particular requirements of the application.

8.7. SESSION

8.7.1. Introduction

This section details the implementation requirements for the Session layer. It is the intent of this section to follow the ISO Session Standards to the fullest extent possible. Where those specifications are inadequate, this section should provide the necessary information.

8.7.2. Session Layer Services

8.7.2.1. Session Functional Units

The following functional units are within the possible scope of an NBS conformant system:

Kernel: This functional unit supports the basic Session services required to establish a Session connection, transfer normal data and release the Session connection. This is a non-negotiable functional unit.

Duplex: This functional unit supports the two way transfer service.

Resynchronize: This functional unit supports the resynchronize service which allows the Session user to set the Session connection to a previous or new synchronization point.

Exceptions: The exceptions functional unit supports the user exception and provider exception reporting services. This functional unit can only be selected when the half-duplex functional unit has been selected.

Activity Management: The activity management functional unit supports the activity management services and the give control service. The major/activity token is available when this functional unit has been selected.

Half-duplex: The half-duplex functional unit supports the half-duplex service. The data token is available when this functional unit has been selected. It is not possible to select both this functional unit and the duplex functional unit for use on the same session connection.

Minor Synchronize: The minor synchronize functional unit supports the minor synchronization point service. The synchronize minor token is available when this functional unit has been selected.

8.7.2.2. Use of Transport Services

The use of Transport layer services by the Session layer functional units defined in section 4.2.1 are as specified in reference [11].

8.7.3. Protocol Agreements

Implementations shall be based on references [9] and [10].

Basic concatenation is only allowed when required by encoding rules. In situations where basic concatenation requires a category 0 SPDU to be combined with a category 2 SPDU, the category 0 SPDU contains neither the Token Item parameter nor user data.

Basic concatenation is required by the session protocol standard. Extended concatenation is not required and can be refused using the normal negotiation mechanisms of the session protocol.

Session segmenting is not required and can be refused using the normal negotiation mechanisms of the session protocol.

Reuse of a transport connection is not required and can be refused.

The use of transport expedited is as stated in the session protocol specification: if available, transport expedited must be used.

The version number shall be used to specify the use of unlimited user data during connection establishment as dictated by ISO/TC97/SC21 N480. The maximum length of user data in the CONNECT SPDU shall be 10K octets; if the length of this user data is no greater than 512 octets a PGI of 193 is used, otherwise a PGI of 194 is used.

> User data will be limited to 10K on all other SPDUs with a PGI of 193 when version 2 of Session has been negotiated, except those SPDUs sent on transport expedited (i.e. the transport expedited limitation is not affected by this change: an ABORT SPDU still has a User Data parameter field that is a maximum of 9 octets).

S-selectors are encoded as a string of octets, the meaning of which is known only to the local system. The length of the string cannot exceed 16 octets. Note: An S-selector has been referred to as an SSAP address in reference [10]. This is a defect and a defect report has been submitted to the ISO Session Rapporteur.

The maximum length of the Reflect Parameter Values parameter in the S-P-Exception-Report SPDU is 1024 octets.

8.8. CONFORMANCE

In order for an implementation to be in conformance with the NBS implementor's agreements, the following rules shall be adhered to:

The required functional units listed in the appropriate section of this implementors' guide shall be included in all NBS conformant systems and shall be implemented in accordance with the agreements in this specification.

This does not mean that all functional units for each of the three layers must be implemented. The particular functional units chosen are determined by the requirements of the entities using the services of ACSE, Presentation, and Session.

A conformant implementation must be ISO conformant as well as meet all of the requirements of this specification. The references listed in section 1.1 of the UL chapter shall be used as the supporting documents for all implementations of ACSE, Presentation, or Session.

Guidelines for implementation of standards' defects will be as per the resolution of such defects by the appropriate ISO standards committee.

The protocol rules for extensibility, as defined in reference [12], shall be followed for ACSE and Presentation.

Note: The status of these rules with respect to Session is undetermined and is being dealt with via defect reports.

8.9. TEST REQUIREMENTS

APPENDIX 8.A: REGISTERED NAMES

Transfer Syntax: NBS-TSI

Transfer Syntax Name:

{ISO registration-authority NBS FTAM ( ) transfer syntax (3) NBS-TSI (0)}

Encoding Rules:

ASN.1 Basic Encoding Rules shall apply.

The first bit of a "mantissa" must be "1".

Transfer Syntax Definition:

The transfer syntax shall be that which results from applying the encoding rules described above to the individual data elements.

APPENDIX 8.B: RECOMMENDED PRACTICES

Reflect Parameter Values

The optional Reflect Parameter Values parameter in the provider ABORT SPDU shall be encoded so as to represent--at the moment a protocol error was detected--the Session connection state, the incoming event, and the first invalid SPDU field.

The first octet encodes the Session state as a number relative to 0 as detailed in Table 1.

The second octet encodes the incoming event as a number relative to 0 as detailed in Table 2.

The third octet contains the SI, PGI, or PI Code of any SI field, PGI unit, or PI unit in error.

NOTE: The remaining 6 octets are undefined herein.

Tbl. 8.1 Session States

State	rel. #	Description
1	0	Idle, no transport connection
1B	1	Wait for T-connect confirm
1C	2	Idle, transport connected
2A	3	Wait for the ACCEPT SPDU
3	4	Wait for the DISCONNECT SPDU
8	5	Wait for the S-CONNECT response
9	6	Wait for the S-RELEASE response
16	7	Wait for the T-DISCONNECT indication
713	8	Data Transfer state
1A	9	Wait for the ABORT ACCEPT SPDU
4A	10	Wait for the MAJOR SYNC ACK SPDU or PREPARE SPDU
4B	11	Wait for the ACTIVITY END ACK SPDU or PREPARE SPDU
5A	12	Wait for the RESYNCHRONIZE ACK SPDU or PREPARE SPDU
5B	13	Wait for the ACTIVITY INTERRUPT SPDU or PREPARE SPDU
5C	14	Wait for the ACTIVITY DISCARD ACK SPDU or PREPARE SPDU
6	15	Wait for the RESYNCHRONIZE SPDU or PREPARE SPDU
10A	16	Wait for the S-SYNC-MAJOR response
10B	17	Wait for the S-ACTIVITY-END response
11A	18	Wait for the S-RESYNCHRONIZE response
11B	19	Wait for the S-ACTIVITY-INTERRUPT response
11C	20	Wait for the S-ACTIVITY-DISCARD response
15A	21	After PREPARE, wait for the MAJOR SYNC ACK SPDU or the ACTIVITY END ACK
15B	22	After PREPARE, wait for the RESYNCHRONIZE SPDU or the ACTIVITY DISCARD SPDU
15C	23	After PREPARE, wait for the RESYNCHRONIZE ACK SPDU, or the ACTIVITY INTERRUPT ACK SPDU or the ACTIVITY DISCARD ACK SPDU
18	24	Wait for GIVE TOKENS ACK SPDU
19	25	Wait for a recovery request or SPDU
20	26	Wait for a recovery SPDU or request
21	27	Wait for the CAPABILITY DATA ACK SPDU
22	28	Wait for the S-CAPABILITY-DATA response

Tbl. 8.2 Incoming Events

Event	rel. #	Description
SCONreq	0	S-CONNECT request
SCONrsp+	1	S-CONNECT accept response
SCONrsp-	2	S-CONNECT reject response
SDTreq	3	S-DATA request
SRELreq	4	S-RELEASE request
SRELrsp+	5	S-RELEASE accept response
SUABreq	6	S-U-ABORT request
TCONcnf	7	T-CONNECT confirmation
TCONind	8	T-CONNECT indication
TDISind	9	T-DISCONNECT indication
TIM	10	Time out
AA	11	ABORT ACCEPT
AB-nr	12	ABORT - no reuse
AC	13	ACCEPT
CN	14	CONNECT
DN	15	DISCONNECT
DT	16	DATA TRANSFER
FN-nr	17	FINISH - no reuse
RF-nr	18	REFUSE - no reuse
SACTDreq	19	S-ACTIVITY-DISCARD request
SACTDrsp	20	S-ACTIVITY-DISCARD response
SACTEreq	21	S-ACTIVITY-END request
SACTErsp	22	S-ACTIVITY-END response
SACTIreq	23	S-ACTIVITY-INTERRUPT request
SACTIrsp	24	S-ACTIVITY-INTERRUPT response
SACTRreq	25	S-ACTIVITY-RESUME request
SACTSreq	26	S-ACTIVITY-START request
SCDreq	27	S-CAPABILITY-DATA request
SCDrsp	28	S-CAPABILITY-DATA response
SCGreq	29	S-CONTROL-GIVE request
SEXreq	30	S-EXPEDITED-DATA request
SGTreq	31	S-TOKEN-GIVE request
SPTreq	32	S-TOKEN-PLEASE request
SRELrsp-	33	S-RELEASE response reject
SRSYNreq(a)	34	S-RESYNCHRONIZE request abandon
SRSYNreq(r)	35	S-RESYNCHRONIZE request restart
SRSYNreq(s)	36	S-RESYNCHRONIZE request set
SRSYNrsp	37	S-RESYNCHRONIZE response
SSYNMreq	38	S-SYNC-MAJOR request
SSYNMrsp	39	S-SYNC-MAJOR response
SSYNmreq	40	S-SYNC-MINOR request
SSYNmrsp	41	S-SYNC-MINOR response
STDreq	42	S-TYPED-DATA request

Incoming Events Cont....

Event	rel. #	Description
SUERreq	43	S-U-EXCEPTION-REPORT request
AB-r	44	ABORT - reuse SPDU
AD	45	ACTIVITY DISCARD SPDU
ADA	46	ACTIVITY DISCARD ACK SPDU
AE	47	ACTIVITY END SPDU
AEA	48	ACTIVITY END ACK SPDU
AI	49	ACTIVITY INTERRUPT SPDU
AIA	50	ACTIVITY INTERRUPT ACK SPDU
AR	51	ACTIVITY RESUME SPDU
AS	52	ACTIVITY START SPDU
CD	53	CAPABILITY DATA SPDU
CDA	54	CAPABILITY DATA ACK SPDU
ED	55	EXCEPTION DATA SPDU
ER	56	EXCEPTION REPORT SPDU
EX	57	EXPEDITED DATA SPDU
FN-r	58	FINISH - reuse SPDU
GT	59	GIVE TOKENS SPDU
GTA	60	GIVE TOKENS ACK SPDU
GTC	61	GIVE TOKENS CONFIRM SPDU
MAA	62	MAJOR SYNC ACK SPDU
MAP	63	MAJOR SYNC POINT SPDU
MIA	64	MAJOR SYNC ACK SPDU
MIP	65	MINOR SYNC POINT SPDU
NF	66	NOT FINISHED SPDU
PR-MAA	67	PREPARE (MAJOR SYNC ACK) SPDU
PR-RA	68	PREPARE (RESYNCHRONIZE ACK) SPDU
PR-RS	69	PREPARE (RESYNCHRONIZE) SPDU
PT	70	PLEASE TOKENS SPDU with Token Item Parameter
RA	71	RESYNCHRONIZE ACK SPDU
RF-r	72	REFUSE - reuse SPDU
RS-a	73	RESYNCHRONIZE - abandon SPDU
RS-r	74	RESYNCHRONIZE - restart SPDU
RS-s	75	RESYNCHRONIZE - set SPDU
TD	76	TYPED DATA SPDU

9. SERVICE ACCESS POINTS AND SELECTORS

The following guidelines on the size of n-selectors apply to implementations based on these agreements and therefore to incoming connection requests only. Outgoing connection requests will support the full range of n-selector sizes.

9.1 UPPER LAYER AGREEMENTS

The following upper layer addressing agreements have been reached.

- The combination of NSAP address, TSAP selector, SSAP selector, PSAP selector and PSAP address must be unique to identify an application entity.

- It is implicitly agreed that the procedure followed for the assignment of NSAP addresses insures that they are globally unique.

- The assignment of TSAP, SSAP, and PSAP selectors is a local end system issue and the values are administered locally.

- SSAP selectors are encoded as a string of octets, the meaning of which is known only to the local system. The length of the string cannot exceed 16 octets.

- PSAP selectors are encoded as a string of octets, the meaning of which is known only to the local system. The length of the string cannot exceed 16 octets.

9.2 TRANSPORT CLASS 4 SERVICE ACCESS POINTS OR SELECTORS

The TSAP selector field in the CR and CC TPDUs shall be encoded as a variable length field and will be interpreted as an octet string. The length of the string cannot exceed 32 octets.

9.3 TRANSPORT CLASS 0 SERVICE ACCESS POINTS

For communicating with public MHS systems, Section 5 of X.410 specifies the use and format of TSAP identifiers.

10. ISO DIS/IS FILE TRANSFER, ACCESS, & MANAGEMENT PROTOCOL

10.1 INTRODUCTION

This section defines Implementors' Agreements based on ISO File Transfer, Access, and Management (FTAM), as defined in ISO DIS 8571. This Draft International Standard has four parts. Part 1 of the DIS gives general concepts, Part 2 defines the Virtual File Store (VFS), Part 3 defines the File Service, and Part 4 defines the File Protocol.

FTAM, as described in the DIS, depends on ISO definitions of ASN.1 (ISO DIS 8824 and 8825), the Presentation Service and Protocol (ISO DIS 8822 and 8823), the Session Service (ISO 8326/CCITT X.215) and Session Protocol (ISO 8327/CCITT X.225), and Transport Class 4. FTAM Phase 2, defined in this section, also requires ACSE Services (ISO DIS 8649/2 and ACSE Protocol (ISO DIS 8650/2). These services and protocols are defined architecturally in the OSI Reference Model (ISO 7498). This section presumes that the reader is familiar with these standards, and possesses technical knowledge appropriate to implementing or testing them. This section provides detailed guidance for the implementor, and is not an FTAM tutorial.

The general agreements reached with respect to the ISO File Transfer, Access, and Management Protocol (FTAM) are:

> FTAM is defined in phases. The Phase 1 FTAM implementation specification (stable document) is based on the second ISO draft proposal, dated April 30, 1985[1], and the ISO draft proposals 8824 and 8825.
>
> The Phase 2 FTAM specification (this section) is based on the Draft International Standard (DIS) and later will be based on the International Standard (IS). THERE IS NO BACKWARD COMPATIBILITY WITH NBS FTAM PHASE 1.

Backward compatibility is impossible, since Phase 1 uses Session services directly, while Phase 2 uses ACSE and Presentation services. Furthermore, there are differences in Filestore, PDU Abstract Syntax, FADU Abstract Syntax, and Transfer Syntax. There also are differences in the Transparency mechanisms and service class negotiations.

---

[1] Part 1 is dated April 20, 1985; Part 2 dated April 29, 1985; and Parts 3 and 4 dated April 30, 1985.

Assuming that Phase 2 FTAM implementations will be based on the forthcoming IS, and that this IS or the ACSE IS provides the ability to pass "user version" information, a mechanism exists for backward compatibility. It is the goal of these agreements to use the "user version" mechanism to provide at least one level of backward compatibility for all future NBS FTAM Phases, facilitating backward compatibility for future FTAM products.

10.2 SCOPE AND FIELD OF APPLICATION

10.3 STATUS

This version of the FTAM implementation agreements was completed March 12, 1987 with respect to the DIS. No further enhancements will be made to this version until the FTAM IS text is available. See the next section, Errata.

10.4 ERRATA

This section shall contain any and all corrections and clarifications, to this version of the agreements that are identified after March 1987. Each change shall be dated. Only text for clarification and correction of errors shall appear here. The correction of typographical errors that do not affect the meaning will not be noted. Text enhancements as a result of DIS-to-IS progression will not be included in this Section.

10.5 ASSUMPTIONS

1. These agreements are based on the ISO 8571 DIS version of FTAM. When the IS text is approved following the close of DIS ballots, the agreements will be modified as necessary to meet the IS specifications.

2. FTAM protocol machines must be able to parse and process up to 7K octets of File PCI and FTAM user data (including grouped FPDUs) as they would be encoded with the ASN.1 Basic Encoding Rules. It is recommended, however, that Presentation user data not be restricted in size.

3. In order to maximize interoperability, it is important that the implementations of FTAM service providers do not unnecessarily restrict the service user's ability to generate arbitrary file service requests. Otherwise, they may not be able to work with FTAM Responders whose operation is constrained by their mapping of the FTAM virtual filestore to their local filestore. For example, error procedures should only be invoked when an error actually occurs, not at the point of the specification of options which might result in a error.

4. Implementations must be able to parse all valid DIS optional parameters if they are present in the PDU. Only those optional parameters specified as mandatory in these agreements are required to be supported for Request and Response PDUs. If these parameters are not present, a default value is assigned locally. A responder should not refuse a request solely because a parameter that is optional in the FTAM standard, but is mandatory in these agreements, is not present.

5. Consideration of any standardized service interface is not covered by these agreements.

10.6 PRESENTATION AGREEMENTS

The following Abstract Syntaxes are supported.

FTAM PCI (including ISO 8571-FTAM and ISO 8571-FADU)
ISO 8650-ACSE1
NBS-AS1
NBS-AS2
NBS-AS3

If the presentation context management functional unit is available, it is possible to use P-ALTER-CONTEXT to negotiate the use of an abstract syntax.

10.7 FTAM SERVICE TYPE AGREEMENTS

The Reliable File Service level (excluding Recovery and RestartDataTransfer functional units) is to be implemented. Implementing the error recovery protocol machine is not required.

10.8 SERVICE CLASS AGREEMENTS

Implementation of the following service classes is defined.

- o File Transfer
- o File Access
- o File Management
- o File Transfer and Management
- o Unconstrained

10.9 FUNCTIONAL UNIT AGREEMENTS

Implementation of the following functional units is defined.

- o Kernel
- o Read
- o Write
- o File Access
- o Limited File Management
- o Enhanced File Management
- o Grouping

10.10 FILE ATTRIBUTE AGREEMENTS

Implementation of the Kernel Group of file attributes is defined. If the optional Storage Group and Security Group are implemented, aspects of their implementation are defined. Implementation of the Private Group is not specified.

Responses to an attribute value request shall always include one of the following:

1. An actual file attribute value.

2. A value indicating that the attribute value is not available at this time. Optionally, a diagnostic may be provided indicating that the attribute is not supported.

3. For the purposes of interworking according to these agreements the <ContentsType> attribute is limited to the <DocumentTypeName> format. The <ConstraintSetName> and <AbstractSyntaxName> form, however, may be used by bilateral agreement and should always be parsed correctly when received.

4. The abstract syntax for the <Attributes> parameter specified in the FTAM DIS can lead to ambiguity in that multiple values for a particular attribute

can be encoded. In order to prevent this ambiguity, it is required that NBS implementations only include a single value for each attribute being encoded. It is also required that each attribute is encoded in the order indicated by the ASN.1 syntax for the <Attributes> parameter.

Note that if multiple values are encoded, it is a local implementation issue for the decoding entity as to which value is returned.

Implementations must also be able to parse an <Attributes> parameter with attributes in random order as per the FTAM DIS.

Mandatory Group

A value for file name and contents type will always be available. Only the Kernel Group of attributes is required.

A minimum range is required for <filename> values as specified in ISO 8571/2 No maximum length or format restrictions apply. A system that does not support multi-component <filename> values or extended <filename> characteristics may reject a request involving such a <filename>. All systems must be able to interpret a <filename> with single component values. Requests using single component <filename> values are responded to using single component <filename> values. Responses to requests involving <filename> values having two or more components are not defined here but may be interpreted via bilateral or other external agreements. Use of <filename> values with multiple components is discouraged.

Optional Groups

If the optional Storage Group of file attributes is implemented, an actual value must be available for the <PermittedActions> attribute.

If the optional Security Group of file attributes is implemented, an actual value must be available for the <AccessControl> attribute.

Implementation of the <Private> Group is not specified.

10.11 DOCUMENT TYPE AGREEMENTS

These document types are defined.

FTAM-3 Unstructured Binary
NBS-2 VARCRLF
NBS-3 8859VARCRLF
NBS-4 TEXT
NBS-5 8859TEXT
NBS-6 SEQUENTIAL
NBS-7 RANDOM
NBS-8 INDEXED
NBS-9 FILE DIRECTORY

Part of our ongoing work is to define, discuss, and propose other file types.

Detailed document type definitions are given in Appendix 10A.

Document type Names:

```
DTN    ::= DTName | DTName params
DTName ::= OBJECT IDENTIFIER
params ::= :param | : param params
param  ::= PrimType | PrimType param

PrimType::= INT n*
          | BIT n2
          | IA5 n1
          | 8859 n1
          | OCT n1
          | UTC
          | GEN
          | NULL
          | BOOL
          | FLOAT n3, n4
          | n5
```

n1 - Maximum number of characters/octets in string.

n2 - Number of bits in string (i.e., nonvarying).

n3 - The minimum number of bits required to be maintained in the mantissa for relative precision.

n4 - Number of bits required to represent the largest unbiased integer exponent in 2's complement.

n5 - Position of key in data unit.

n* - Number of octets required to represent, in 2's complement format, the largest integer to be passed.

Note: A document type name may carry one or more parameters separated by ":". A parameter consists of one or more primitive data types each with a number giving its maximum range.

The primitive data types and minimal size range that an implementation must accept for storage are given in Table 10.1.

Tbl. 10.1 FTAM primitive data types

PRIMITIVE DATA TYPE	REPRESENTATION IN PARAMETER	MINIMUM RANGE (OCTETS)
ASN.1 INTEGER	INT n*	(1 - 2)
ASN.1 Bit String	BIT n2	(0 - 1)
ASN.1 IA5String	IA5 n1	(0 - 134)
NBS-AS1 8859String	8859 n1	(0 - 134)
ASN.1 OCTETSTRING	OCT n1	(0 - 512)
ASN.1 BOOLEAN	BOOL	
ASN.1 NULL	NULL	
ASN.1 Generalized Time	GEN	
ASN.1 Universal Time	UTC	
NBS-AS2 Floating Point	FLOAT n3,n4	

Note: The primitive data types and their maximum ranges for a specific file as described by the parameters above are maintained in the contents type file attribute. The contents type file attribute value is established at the file's creation and cannot be changed via FTAM for the life of the file. This implies that the data element types and ranges and data unit formats are fixed for all accessors of that file as long as the file exists.

An <object identifier> is a string of integers; FTAM <document type> parameterization is achieved by exploiting that structure.

The final registration authority entity is followed by a <data unit> description. The <data unit> description is a series of data element descriptions. Each <data element> description is an integer identical to the ASN.1 type code, followed by any required parameter values, as integers.

The following values correspond to the NBS primitives not found in ASN.1 and the integer value for ":" (separator).

FLOAT - 127
8859 - 126
: - 125

The following is an example of the encoding of the NBS-8 document type for an indexed file in which the key is contained in the data unit. The file consists of three fields (2-octet integer, 4-octet integer, and 10-character IA5String) with the key being the second field.

This file is defined in the DocumentTypeName grammar as:

NBS-8: INT 2 INT 4 IA5 10 : INT 4 : 2

By applying the encoding rules given above, we have:

```
  x - encoding for NBS-8 document type name
125 - colon delimiter
  2 - type of first field = integer
  2 - maximum length of first field = 2
  2 - type of second field = integer
  4 - maximum length of second field = 4
 22 - type of third field = IA5String
 10 - maximum length of third field = 10
125 - colon delimiter
  2 - type of key = integer
  4 - maximum length of key = 4
125 - colon delimiter
  2 - position of key = second field
```

The following is an example of the encoding of the NBS-8 document type for an indexed file in which the key is not contained within the data unit. The file consists of four fields (12-character IA5String, 4-octet integer, 2-octet integer and a 3-character IA5String) and a 3-octet integer key which is not contained within the data unit.

This file is defined in the DocumentTypeName grammar as:

```
NBS-8:  IA5 12 INT 4  INT 2  IA5 3  : INT 3  : 0
```

By applying the encoding rules given above, we have:

```
  x - encoding for NBS-8 document type name
125 - colon delimiter
 22 - type of first field = IA5String
 12 - maximum length of first field = 12
  2 - type of second field = integer
  4 - maximum length of second field = 4
  2 - type of third field = integer
  2 - maximum length of third field = 2
 22 - type of fourth field = IA5String
  3 - maximum length of fourth field = 3
125 - colon delimiter
  2 - type of key = integer
  3 - maximum length of key = 3
125 - colon delimiter
  0 - position of key = 0 (outside of DU)
```

The following notation allows the transfer of floating point numbers, while retaining their meaning, as defined by existing standards IEC 559 and IEEE 754.

```
FloatingPointNumber  ::= [PRIVATE 0]   CHOICE{
        finite       [0]   IMPLICIT SEQUENCE{
                Sign,
                mantissa BIT STRING,
```

```
                exponent    INTEGER},
        infinity            [1] IMPLICIT Sign,
        signallingNaN       [2] IMPLICIT NaN,
        quiteNaN            [3] IMPLICIT NaN,
        zero                [4] IMPLICIT NULL}
Sign                        ::=       INTEGER{
        positive            (0),
        negative            (1)}
NaN                         ::=       INTEGER
```

Notes: 1. The mantissa is a number in the range ($1/2 \leq$ mantissa <1).
2. The value is equal to mantissa * 2^{exponent}.
3. The first bit in the mantissa is most significant.
4. See IEEE 754 for definitions of terminology, such as NaN.

10.11.1 Character Sets

IA5 and 8859/1 character sets have been specified, and are to be implemented as described below.

1. IA5

The IA5 character set leaves 2 options and 10 characters unspecified. The definitions used are:

2/3	#
2/4	$
4/0	@
5/11	[
5/12	\
5/13	]
5/14	^
6/0	'
7/11	{
7/12	¦
7/13	}
7/14	~

Note: This is exactly the International Reference Version (IRV) specified in the IA5 standard except that the code 2/4 has the graphic rendition "$" instead of the IRV-specified value of the International Currency Symbol.

Control characters should be handled as follows.

a. Semantics of format effectors will be preserved.

b. Transmission control characters, device control characters, information separators, and "other" control characters are simply preserved via their codes.

c. Code extension shall not be used. If it is received, the code extension characters should be preserved, as in the case of the transmission control characters, and any printing characters that form later parts of escape sequences are interpreted as stand alone characters.

d. Combined horizontal and vertical movement of cursor positioning is not to be preserved.

2. 8859/1

The Latin Alphabet No. 1 is used to specify the printable rendition of C0 and C1. C0 control characters and their associated rules are taken from the IA5 definition. C1 control characters simply have their codes preserved across a transfer.

10.11.2 Document Type Negotiation Rules

1. Connection Establishment

In Connection Establishment, <DocumentTypeName> values are negotiated by subset of the proposed base set of <DocumentTypeName> values, without regard to DU syntax parameter(s) that may be supplied on any <DocumentTypeName> that requires a DU syntax specification.

2. File Creation

An F-CREATE Request FPDU must contain a <DocumentTypeName> value from the negotiated set of base <DocumentTypeName> values. If the <DocumentTypeName> used requires DU syntax parameters, then these parameters must be supplied. If the <DocumentTypeName> used requires DU syntax parameters and none are provided on the F-CREATE Request, then the F-CREATE Request FPDU must be rejected.

3. File Opening

The <DocumentTypeName> form (optionally with appropriate DU syntax parameters) should always be used when proposing a <ContentsType>.

Similarly, an F-OPEN response should use the <DocumentTypeName> option (with appropriate DU syntax parameters) in the <Contents Type> field. This allows the receiving entity to use the <DocumentTypeName> attributed to the file instead of receiving a <Constraint Set Name> and <Abstract Syntax Name> pair, which does not reflect the file information contained in the NBS document types.

Note: An F-OPEN response without a <DocumentTypeName> (but carrying the <Constraint Set Name> and <Abstract Syntax Name> form) may cause the initiator to issue an F-CLOSE request.

10.11.3 Relationship Between DUs, DEs and Document Types

"Abstract Syntax" is used to refer to the syntactic information which is architecturally passed between the Application and Presentation Layers. The Abstract Syntax defines Data Element (DE) types which are not necessarily ASN.1 primitive types. A Data Element (DE) is the smallest piece of data whose identity is necessarily preserved by the Presentation Service. Data types may be made up of other data types. Data Elements are not defined in terms of other Data Elements.

A <data unit> (DU) is a sequence of one or more data elements. Architecturally, entire, single DEs are passed into and out of the application process. In a real implementation, DUs may be passed.

To maintain DU boundaries during transfer, file structuring information must be passed (ISO 8571-FADU DEFINITIONS, FTAM Part 2 Section 5.3.2). A data element is referred to as a File Contents Data Element in ISO 8571-FADU DEFINITIONS.

Document types refer to aspects of local processing and storage. They describe:

- o structural relationship between DUs,
- o structure of DUs, called DU syntax, and
- o data element types found in the file.

Because document types pertain to local processing and storage, the DU syntax makes assertions about the syntax and the size of DUs (records) in storage. Parameters on the document types provide this information about the syntax and size of the DUs.

10.12 F-CANCEL ACTION

When an F-CANCEL is sent or received, the following occurs:

- o no more data is sent,
- o <checkPointNumbers> are removed, and
- o state of the file is implementation dependent.

10.13 DIAGNOSTIC AGREEMENTS

1. A <diagnostic> parameter is mandatory only when the Action Result or State Result is not zero. (The nature of these agreements is to provide <diagnostic> information when any result parameter is not <success>.)

2. General catch-all diagnostic action is discouraged.

3. A <furtherDetails> subfield is mandatory. Use of octet string is discouraged.

4. Use of F-P-ABORT for other than protocol errors and catastrophic situations is discouraged.

5. When returning an error status in a file management related diagnostic (i.e., F-READ-ATTRIBUTEresponse or F-CHANGE-ATTRIBUTEresponse), identify the erroneous attribute by using the first two characters of <further-details> to hold a 2-digit number (encoded in IA5String) from the F-READ-ATTRIBUTErequest attributes abstract syntax definition (ISO/DIS 8571/4 Section 20-4):

00	Filename
01	Contents-Type
02	Storage Account
03	Date and Time of Creation
04	Date and Time of Last Modification
05	Date and Time of Last Read Access
06	Date and Time of Last Attribute Modification
07	Identity of Creator
08	Identity of Last Modifier
09	Identity of Last Reader
10	Identity of Last Attribute Modifier
11	File Availability
12	Permitted Actions
13	Filesize
14	Future Filesize
15	Access Control
16	Encryption Name
17	Legal Qualifications
18	Private Use

6. The set of File Management <diagnostics>, found in Table 44 of ISO 8571/3 Annex A, must be maintained.

7. The <diagnostic> parameter values defined in FTAM (8571/3 Annex A) are partitioned into sets that apply to:

   a. general FTAM <diagnostics> (all but identification = 1 are recommended against),
   b. Protocol and supporting services,
   c. the <FTAM Regime>,
   d. the <File Selection Regime>,
   e. the <File Open Regime>,
   f. the <Data Transfer Regime>.

Each of those sets is further partitioned into <diagnostics> applicable to each parameter of the corresponding service elements.

In the case where a specific parameter can in no way be accommodated then the request fails and a <diagnostic> indicating one such parameter should be returned by the responder. In the case where a negotiable parameter cannot be

accommodated with exactly the value requested but is negotiated to a different value (as defined in Section 10.17) then the request formally succeeds but informative <diagnostics> indicating those parameters negotiated should be returned.

10.14 CONCURRENCY

The <concurrency control> used by default on the file selection and file open regime for the first file accessor of a file is:

read	shared
insert	exclusive
replace	exclusive
extend	exclusive
erase	exclusive
rattr	shared
cattr	exclusive
del file	exclusive

10.15 REQUESTED ACCESS

The <RequestedAccess> parameter on <F-SELECT> or <F-CREATE> is used to specify the actions which the initiator may perform during the file selection. The value of the <RequestedAccess> parameter is compared by the responder to the <AccessControl> and <PermittedActions> file attributes and concurrency controls (including those requested by the initiator) currently in place on the file. If the value of the <RequestedAccess> parameter is not consistent with either <AccessControl>, <PermittedActions>, or concurrency controls in place, then the <F-SELECT> or <F-CREATE> must be rejected.

<RequestedAccess> is consistent with <AccessControl> if, for each action requested, that action either requires no password, or the required password has been specified on the <F-SELECT> or <F-CREATE> request.

<RequestedAccess> is consistent with <PermittedActions> if, for each action requested, that action is allowed by the <PermittedActions> file attribute.

<RequestedAccess> is consistent with <ConcurrencyControl> requested on the <F-SELECT> or <F-CREATE> if, for each action requested, that action has not been specified as <not required> or <not allowed> in the <ConcurrencyControl> parameter.

<RequestedAccess> is consistent with concurrency controls in place on the file if for each action requested no other accessor of the file has set the concurrency control for that action to either <exclusive> or <not allowed>.

10.16 SECURITY

10.16.1 Optional Password Support

Users may provide values for <InitiatorIdentity> and <FilestorePassword>.

Password support in FTAM is not required. If this information is provided, it will be sent to the Responder on the F-INITIALIZE.

The syntax of <InitiatorIdentity> and <FilestorePassword> is system-dependent. <InitiatorIdentity> and <FilestorePassword> will represent "account" information on the local system, which may be different from the <account> parameter.

10.16.2 Access Passwords

Users may provide <accessPasswords>. If the information is provided, the passwords will be sent to the Responder in the <accessPasswords> parameter.

10.16.3 Anonymous User Convention

A commonly defined "anonymous user" convention is to be provided for all systems that choose to support this capability. The access available to that user is locally determined. The <InitiatorIdentity> value to be used is ANON. Any password should succeed.

10.16.4 Implementation Responsibilities

It is the responsibility of each local system to provide security for its own real filestore. Encryption of passwords will not be done by FTAM.

A user of the file service must be known by the Responder. "Known" is defined by the local Filestore, and is dependent on the level of security provided by the local Filestore.

10.17 NEGOTIATION

The guidelines for negotiation that have been agreed upon are outlined in Table 10.2.

Tbl. 10.2 FTAM negotiation rules

Service or Parameter	Depends on or may be negotiated down by:
F-INITIALIZE	
Req.Pres.ContextMgmt	Success or failure.
Req.Func. Unit	Optional functional units, negotiated by subset (as per DIS 8571/3 Section 10.3). (Affects session functional units.)
Req.Attr. Groups	Negotiated by subset.
Req.Comm.Quality of services	Reference session.
Req.ContentTypeList	Negotiated by subset.
F-SELECT	
Attributes	Only by <filename>. N.B. The <filename> on response and confirm must be that of an existing virtual file.
Requested Access	Negotiated by subset (in case of complete or partial success), must be consistent with Functional Units negotiated, access control attribute, and permitted actions attribute.
F-CREATE	Consistent with Functional Units.
Initial attributes	1. The attributes returned are within the subset negotiated at initialization. 2. The individual attribute values returned must be consistent with negotiation/ranges for that attribute 3. The Responder returns values for all attributes which differ from the actual request.
Requested Access	As in F-SELECT.
F-DELETE	Consistent with Functional Units.
F-READ-ATTR	Consistent with Functional Units, service class, and requested access.
F-CHANGE-ATTR	As for F-READ-ATTRIBUTE.

(Continued on next page.)

Tbl. 10.2 FTAM negotiation rules, continued

Attributes	If any attribute cannot be successfully changed, then an error more severe than warning should be returned and no attribute should be changed.
F-OPEN	
Processing Mode	Not Negotiated. Must be consistent with Functional Unit and Requested Access negotiated, with the permitted actions attribute, and with the contents type name.
Contents Type	As defined in the DIS.
Concurrency Control	More restrictive than the concurrency control of F-SELECT; consistent with the concurrency control of other users.
F-BEGIN-GROUP	Consistent with Functional Units and F-END-GROUP service class.
F-LOCATE	Consistent with Functional Units, F-ERASE, requested access (and therefore with permitted actions) Service Class and Processing Mode.
F-READ F-WRITE	Functional Units, Service Class and number of bulk data transfers, Requested Access and Processing Mode.
F-DATA/F-DATA-END F-CANCEL F-TRANSFER-END	Functional Units and Service Class.

10.18 REQUIREMENT FOR CONFORMANT IMPLEMENTATIONS

This section gives the criteria to be satisfied by every implementation of FTAM that conforms to these agreements.

Conformance to these agreements is stated in terms of the different roles

occupied by FTAM implementations. The interoperability of certain configurations of these roles motivates this approach. Interoperable configurations of these roles are given in Section 10.18.1.

The only function provided by every conformant implementation is the transfer of unstructured binary files in their entirety. It must be recognized that such simple transfer, while commonly understood and generally important, will not support all application of FTAM. Section 10.19 defines implementation classes of FTAM services and protocol that can provide other specific functions. Those other functions exploit the access and management capabilities of FTAM. The unconstrained service class (with appropriately chosen functional units) can be used to provide the functions of any of the implementation classes. Users of FTAM must consider carefully what functions they require. They must examine all the implementation classes and select according to their needs.

10.18.1 Interoperable Configurations

Any implementation conforming to this specification must be able to act in at least one of the following role combinations:

1. initiator and receiver,

2. initiator and sender,

3. responder and sender,

4. responder and receiver.

Minimal implementations of combination 1 will interoperate with minimal implementations of combination 3. Minimal implementations of combination 2 will interoperate with minimal implementations of combination 4.

Any implementations of roles 1 and 3 will be able to interoperate at the intersection of their capabilities (which will be at least the minimal capabilities described in Sections 10.18.3 to 10.18.8). Any implementations of roles 2 and 4 will be able to interoperate at the intersection of their capabilities (which will be at least the minimal capabilities described in Sections 10.18.3 to 10.18.8).

These role combinations and this interoperability are shown in the table below.

Tbl. 10.3 Interoperable configurations

		Initiator		Responder	
		sender	receiver	sender	receiver
Initiator	sender				x
	receiver			x	
Responder	sender		x		
	receiver	x			

10.18.2 Relationship to ISO 8571--The FTAM Standard

Any implementation in conformance to ISO 8571 (as defined in ISO 8571/4 Section 21 (Conformance)), in addition to the implementation of the minimal protocols and roles enumerated in Sections 10.18.3 to 10.18.8, is considered to be in conformance with these agreements. Any implementation violating any of the conformance statements in ISO 8571/4 is considered to be in violation of these agreements.

10.18.3 Requirements for Document Type Support

The document type FTAM-3 shall be supported for purposes of transfer and storage. The details regarding support for FTAM-3 in the FTAM dialogue are given in Sections 10.18.6 and 10.18.7.

Support of document types other than FTAM-3 (including document types defined elsewhere in these agreements or registered by authorities other than the NBS) is not required by these agreements. However, for the support for any document type these agreements require that the mechanisms for referencing those document types in the protocol be consistent with the protocol in both state and encoding. Support for document types described in these agreements also entails support for :

- o the semantics given in their description,
- o the preferred transfer syntax (via the designated transfer syntax name and the name "{ISO standard 8825}," and
- o the transfer of that document type under access context US (unstructured).

10.18.4 Initiators

Every implementation of an FTAM initiator shall support:

- o the kernel protocol and its mandatory parameters with minimum ranges [Minimum required ranges are specified in Section 10.18.8.],
- o the grouping protocol and the threshold parameter with a value of 2 for use in the file transfer class,
- o at least one of the read or write protocols [Specific conformance for reading and writing is defined in Sections 10.18.6 and 10.18.7.],

and support the applicable procedures defined in ISO 8571/4 Sections 8.1 (FTAM regime establishment), 8.2 (FTAM regime termination), 8.3 (File selection), 8.4 (File deselection), 8.9 (File open), 8.10 (File close), 8.11 (Begin group), 8.12 (End group), and 10 (File general actions). To support the above protocols and procedures the implementation shall be able to:

- o request the kernel, grouping and at least one of the read or write functional units,
- o request the <reliable file service> (but not necessarily any error control procedures) with the "service level" parameter,
- o request the file transfer class with the "service class" parameter,
- o request optional functional units consistently from those defined for the file transfer service class,
- o request the document type FTAM-3 using the "document type name" form of the contents type parameter, and
- o request a "communication quality of service" consistent with the transport definition in these agreements

as part of the filestore initialization procedures in ISO 8571/4 Section 8.1, FTAM regime establishment.

Initiators must be able to operate under all circumstances if the above minimum values are successfully negotiated and returned on an F-INITIALIZEresponse PDU. Initiators must be able to operate with any downward negotiation of requested parameter values as described in Section 10.17.

Should the supporting services break down, such that FTAM communication is impossible, the FTAM protocol machine shall notify the user with an <F-P-ABORT indication> and <diagnostic> with identifier 1011, as well as any known "further details".

10.18.5 Responders

Every implementation of an FTAM responder shall support:

- the kernel protocol and its mandatory parameters with minimum ranges [Minimum required ranges are specified in Section 10.18.8.],
- the grouping protocol and the threshold parameter with a value of 2 for use in the file transfer class,
- at least one of the read or write protocols [Specific conformance for reading and writing is defined in Sections 10.18.6 and 10.18.7],

and support the applicable procedures, defined in ISO 8571/4 Sections 9.1 (FTAM regime establishment), 9.2 (FTAM regime termination), 9.3 (File selection), 9.4 (File deselection), 9.9 (File open), 9.10 (File close), 9.11 (Begin group), 9.12 (End group), and 10 (File general actions). To support the above protocols and procedures the implementation shall be able to:

- accept requests for the kernel, grouping and at least one of the read or write functional units,
- accept requests for the <Reliable File Service> (but not necessarily any error control procedures) with the "service level" parameter,
- accept requests for the file transfer class with the "service class" parameter,
- accept requests for optional functional units consistently from those defined for the file transfer service class and consistent with its supported roles as specified in Section 10.17,
- accept the document type FTAM-3 using the "document type name" form of the contents type parameter, and
- accept requests for a "communication quality of service" consistent with the transport definition in these agreements

as part of the filestore initialization procedures in ISO 8571/4 Section 9.1, FTAM regime establishment.

Responders must be able to operate under all circumstances if the above minimum values are requested on an F-INITIALIZErequest PDU. Responders must not negotiate upward in the sense described in Section 10.17.

Each responder shall support, for purposes of both communication and processing, the kernel and storage groups of attributes. A value shall always be available for the kernel attributes. For the storage attributes a value of

"no value available" may be the only applicable value.

Responders must complete each action requested and supported in a manner consistent with its description in ISO 8571/2 Sections 6 (Actions on complete files) and 7 (Actions for file access), and must interpret each supported attribute in a manner consistent with its definition in ISO 8571/2 Section 8 (File attributes).
Under circumstances where actions cannot be carried out either as requested or consistently with ISO 8571/2 Sections 6 (Actions on complete files) and 7 (Actions for file access), the responder must return at least one diagnostic indicating:

- o if the failure was due to either a protocol or filestore failure, and then:

  -- precisely which action failed,
  -- at least one of the parameters that could not be accommodated with the diagnostic type indicating at least the degree of failure, as given by the action and state result parameter, or

- o that the failure was due to unforeseen system shutdown.

Should the supporting services break down, such that FTAM communication is impossible, the FTAM protocol machine shall notify the user with an <F-P-ABORT indication> and <diagnostic> with identifier 1011, as well as inform the user of any known "further details".

10.18.6 Senders

Every implementation of an FTAM sender shall support the read functional unit as responder or the write functional unit as initiator, and support the applicable procedures defined in ISO 8571/4 Sections 11 (State of the bulk data transfer activity), 12 (Bulk data transfer protocol data units), 15 (Bulk data transfer sending entity actions), 17.1 (Discarding), and 17.2 (Cancel).

To support those procedures the implementation shall be able to send files of the NBS document type FTAM-3 and shall be able to send them as user data in PPDUs in blocks of no more than 7168 octets.

10.18.6.1 Initiator Senders

Every implementation of an FTAM sender which is also an FTAM initiator shall support:

- o the write functional unit and protocol, and

- o for the document type FTAM-3 the following bulk data transfer specification parameter:

FADU operation	replace
FADU identity	first
Access context	US

and support the applicable procedures, defined in ISO 8571/4 Section 13 (Bulk data transfer initiating entity actions).

10.18.6.2 Responder Senders

Every implementation of an FTAM sender which is also an FTAM responder shall support:

- the read functional unit and protocol, and
- for the document type FTAM-3 the following bulk data transfer specification parameter:

  FADU identity first
  Access context US

and support the applicable procedures, defined in ISO 8571/4 section 14 (Bulk data transfer responding entity actions).

10.18.7 Receivers

Every implementation of an FTAM receiver shall support the read functional unit as initiator or the write functional unit as responder, and support the applicable procedures, defined in ISO 8571/4 sections 11 (State of the bulk data transfer activity), 12 (Bulk data transfer protocol data units), 16 (Bulk data transfer receiving entity actions), 17.1 (Discarding), and 17.2 (Cancel).

To support those procedures the implementation shall be able to receive files of the NBS document type FTAM-3 and shall be able to receive them as user data in PPDUs in blocks of at least 7168 octets.

10.18.7.1 Initiator Receivers

Every implementation of an FTAM receiver which is also an FTAM initiator shall support:

- the read functional unit and protocol, and
- for the document type FTAM-3 the following bulk data transfer specification parameter:

  FADU identity first
  Access context US

and support the applicable procedures, defined in ISO 8571/4 section 13 (Bulk data transfer initiating entity actions).

10.18.7.2 Responder Receivers

Every implementation of an FTAM receiver which is also an FTAM responder shall support:

- o the write functional unit and protocol, and

- o for the document type FTAM-3 the following bulk data transfer specification parameter:

FADU operation	replace
FADU identity	first
Access context	US

and support the applicable procedures, defined in ISO 8571/4 section 14 (Bulk data transfer responding entity actions).

10.18.8 Minimum Ranges

Any implementation of any conformant FTAM configuration shall be able to receive and meaningfully process all mandatory parameters for all functional units supported as well as the diagnostic parameter within at least the minimum ranges of values given in Table 10.4. A conforming implementation may support a wider range of values for any parameter.

Tbl. 10.4 Required minimal parameter support

Parameter	Minimum Range
diagnostic	Values as specified in ISO 8571/3 Annex A (Diagnostic parameter values) Tables 42, 43 and 45 which correspond directly to mandatory parameters.
action result	All values.
state result	All values.
F_INITIALIZE	
functional units[1]	"read" (for initiator/receivers and responder/senders) or "write" (for initiator/senders and responder/receivers).
presentation context management[2]	"Not required."
all others	As specified in Sections 10.18.4 and 10.18.5 above.
F_SELECT	
attributes	Only filename is used with a minimum supportable length of 8 characters. Any other attribute supported for other services must have minimum supported lengths as in ISO 8571/2 Section 11 (Minimum attribute ranges) Table 10.
requested access	"read" for initiator receivers "read" for responder senders "replace" for initiator senders "replace" for responder receivers
F_OPEN	
processing mode	"read" for initiator receivers "read" for responder senders "replace" for initiator senders "replace" for responder receivers
content type	"FTAM-3"

(Continued on next page.)

Tbl. 10.4 Required minimal parameter support, continued

Parameter		Minimum Range
F_READ		
	FADU identity	"first"
	access context	"US"
F_WRITE		
	FADU operation	"read" for initiator receivers "read" for responder senders "replace" for initiator senders "replace" for responder receivers
	FADU identity	"first"
	access context	"US"
F_BEGIN_GROUP		
	threshold[3]	For file transfer (a minimal required function)2.

---

1 The parameters, functional units, and presentation context management are not ordered, so "minimum value" cannot be formally defined. The above values are those required for conformance to these agreements but no value conformant to ISO 8571 for use in other applications is regarded to be in violation of these agreements.

2. Other functional units (and service classes) for defined implementations may also be valid provided that they are implemented in accordance with these agreements, specifically section 10.18.8.

3 Every implementation must support the threshold value 2 to provide the basic required function of file transfer; any other value in other applications is acceptable.

For any other supported parameters, minimum ranges are taken from the minimum ranges for the attribute corresponding to each as in ISO 8571/2 Table 2.

10.19 IMPLEMENTATION CLASSES

This section defines implementation classes for the specific functions of:

- File Transfer
- File Access
- File Management.

Those definitions are expressed in terms of:

- o Document Types
- o Attributes
- o Service Classes (both service elements and their parameters).

This by no means defines all possible implementation classes.

The following implementation classes (profiles) are defined:

T1: Simple File Transfer
T2: Positional File Transfer
T3: Full File Transfer
A1: File Access
M1: Management.

The following service classes are defined individually for implementations. Note that no defined implementation is precluded from supporting more than one service class.

- o File Transfer
- o File Access
- o File management
- o Unconstrained
- o File Transfer and Management

Support of an implementation class requires adherence to: 1. corresponding definition in DIS 8571/3 Section 8 and any related procedures in DIS 8571/4 Sections 8-17, 2. requirements given in Sections 10.5-10.18 of these agreements, and 3. requirements for parameter and attribute support as defined in Section 10.18.8.

10.19.1 General Requirements for the Defined Implementation Classes

- o Implementations will support the Reliable Service level.

- o Implementations will be able to act either as initiators or responders or both.

- o Implementations that support either the limited file management or both limited file management and enhanced file management must support both the Storage and Security attribute groups.

- o Implementations must support diagnostics as described in Section 10.13 of these agreements.

- o Implementations that support the file access service class will support access to sequential files. Support of sequential files entails hierarchy of depth and arc length = 1. Other hierarchy depth and arc lengths are not precluded by these agreements.

10.19.2 Use of Lower Layer Services

- Support for the Presentation Context Management functional unit is not required.

- Implementations will support the Session, Presentation, and ACSE requirements as stated in Section 8.

Note: Implementation of the Session Resynchronize functional unit is highly recommended, since the F-CANCEL service may be less effective when mapped to S-DATA.

10.19.3 Document Type Requirements for the Defined Implementation Classes

Implementations conformant to implementation classes defined in Table 10.5 will support the following document types with the caveats and procedures given. Those document types are defined in Appendix 10A of these agreements.

- NBS-2
  Caveat: NBS-2 is included only to allow compatibility with file systems storing Phase 1 files.

- NBS-3

- NBS-4

- NBS-6

- NBS-7

  Note: Support of this document type entails the naming of FADUs by their position in preorder traversal.

  Caveat: Other methods of naming FADUs depend on the system, application, and specific file, and as such are not described here.

- NBS-5

  Note: Support for 8859 strings and their interpretation as defined in Section 10.11.1 of these agreements.

- NBS-8

  Note: Document type NBS-9 is optional in all implementation classes.

Support of NBS-2,3,4,5 require the ability for transfer or access using transfer syntax TS-1.

Support for any document type requires the ability to transfer and store the abstract syntax given in its definition. These agreements do not specify techniques or formats for storage.

Caveat: Specific abstract syntaxes for the parameterized document types NBS-6,7,8 are not specified in these agreements.

Any document type supported must be identifiable by its document type name as given in Appendix 10A and, where defined, the parameterization scheme given in Section 10.11 of these agreements.

10.19.4 Parameters for the Defined Implementation Classes

- Use of CCR parameters is not defined within the scope of these agreements.

- Implementations will support the contents type list parameter on the F-INITIALIZE service element. The initiating service must supply a value for this parameter.

- Implementations will support the Diagnostic Parameter as stated in Section 10.13 of these agreements.

- Implementations will support Identity of Initiator Parameter on the F-INITIALIZE Service Element. If the initiating service user supplies a value for this parameter, it will be sent on the request PDU and the virtual filestore will process the parameter. Use must be consistent with Section 10.16 of these agreements.

- Implementations are not precluded from using other parameters for Security and/or accounting.

10.19.5 Parameter Ranges for the Defined Implementation Classes

Parameter ranges for implementations classes are as stated for primitive data types in Section 10.11 of these agreements.

10.19.6 File Attribute Support for Implementations

Implementations of the implementation classes will support file attributes in the following ways.

- If an attribute is "supported" it implies a value will be returned other than the value "no value available," and the value will follow the rules as stated in these agreements and in FTAM 8571 Part 2.

- If an attribute is "optionally supported" a value of "no value available" may be returned.

- If an attribute group is "not supported" then no value will be returned for any of its attributes.

Kernel Group supported

1. Filename	supported
2. Contents Type	supported
Storage Group	supported
1. storage account	optionally supported
2. date and time of creation	optionally supported
3. date and time of last modification	optionally supported
4. date and time of last read access	optionally supported
5. date and time of last attribute modification	optionally supported
6. Identity of Creator	optionally supported
7. Identity of Last Modifier	optionally supported
8. Identity of Last Reader	optionally supported
9. Identity of Last Attribute Modifier	optionally supported
10. File Availability	supported
11. Permitted actions	supported
12. Filesize	supported
13. Future Filesize	optionally supported
Security Group	optionally supported
1. Access Control	supported
2. Encryption Name	optionally supported
3. Legal Qualifications	optionally supported
Private Group	not supported

Tbl. 10.5 Implementation class support requirements

FU	Service Class T	M	A	T&M	UNCST
Kernel	T1,T2,T3	M1	A1,A2		
Read (See note 3.)	T1,T2,T3		A1,A2		
Write (See note 3.)	T1,T2,T3		A1,A2		
Limited File Mgmnt.	SeeNote 6	M1	SeeNote 6	S	S
Enhanced File Mgmnt.		M1			
Grouping	T1,T2,T3	M1		E	E
File Access			A1,A2		
				E	E
Document Types					
FTAM-3	T1,T2,T3		A1,A2	N	N
NBS-2	T1,T2,T3		A1,A2		
NBS-3	T1,T2,T3		A1,A2	O	O
NBS-4	T2,T3		A1,A2		
				T	T
NBS-5	T2,T3		A1,A2		
				E	E
NBS-6	T2,T3		A1,A2		
				4	5
NBS-7	T2,T3		A1,A2		
NBS-8	T3		A2		

Notes:

1. The File Management class is only to be implemented in conjunction with one of the Transfer or Access classes.

2. Class T2 is subset of T3. A1 and T1 are subsets of A2 and T2, respectively.

3. Classes T1, T2, and T3 require the support of read and/or write functional units.

4. The 'file transfer and management' service class may be supported by combining M1 with any of the T - implementation classes.

5. For the 'unconstrained' service class no specific requirement for the selection of functional units is defined. The unconstrained service class can be applied to provide the functions of any of the implementation classes.

6. Limited File Management is not required for the T- and A- implementation classes, but very often it will be a user request to have limited file management functionality available together with file transfer and file access functions. So limited file management may be added as an option to the T- and A- implementation classes.

10.20 PROVISION OF SPECIFIC FUNCTION

10.20.1 Implementation Class T1: Simple File Transfer

Implementation class T1 provides the function of transferring entire files at the reliable file service level for files with an unstructured constraint set. This includes support of the document types.

- o FTAM-3 (unstructured binary)
- o NBS-2 (unstructured text files with embedded <cr><lf> pairs)
- o NBS-3

This implementation class supports file transfer and not file access, that is, the ability to:

- o read a complete file

and/or

- o write (replace, extend) to a file.

10.20.2 Implementation Class T2: Positional File Transfer

Implementation class T2 provides the function of transferring files at the reliable file service level for files with an unstructured or flat constraint set. This includes support of the document types:

- o FTAM-3 (unstructured binary)
- o NBS-2 (unstructured text files)
- o NBS-3 (unstructured text files)
- o NBS-4 (sequential text files)
- o NBS-5 (sequential text files)
- o NBS-6 (sequential files)
- o NBS-7 (random access files)

This implementation class supports file transfer and not file access, that is, the ability to:

- o read a complete file or a single FADU which is identified by key or by position

and/or

- o write (replace, extend, insert) to a file or an FADU.

This implementation class is upward compatible to T1 for the transfer of unstructured files.

10.20.3 Implementation Class T3: Full File Transfer

Implementation class T3 provides the function of transferring files at the reliable file service level for files with an unstructured, flat or general hierarchical constraint set. This includes support of the document types:

- o FTAM-3 (unstructured binary files)
- o NBS-2 (unstructured text files)
- o NBS-3 (unstructured text files)
- o NBS-4 (sequential text files)
- o NBS-5 (sequential text files)
- o NBS-6 (sequential files)
- o NBS-7 (random access files)
- o NBS-8 (indexed sequential files)

This implementation class supports file transfer and not file access, that is, the ability to:

- o read a complete file or a single FADU which is identified by key or by position

and/or

- o write (replace, extend, insert) to a file or an FADU.

This implementation class is upwardly compatible to T1 for the transfer of unstructured files.

10.20.4 Implementation Class A1: File Access

Implementation class A1 provides the function of transfer of and access to files with unstructured or flat constraint sets at the reliable file service level. This includes support of the document types:

- o FTAM-3 (unstructured binary files)
- o NBS-2 (unstructured text files)
- o NBS-3 (unstructured text files)
- o NBS-4 (sequential text files)
- o NBS-5 (sequential text files)
- o NBS-6 (sequential files)
- o NBS-7 (random access files)

This implementation class supports file transfer and file access, that is the ability to:

- o read a complete file or FADUs which are identified by key or by position,
- o write (replace, extend, insert) to a file or an FADU,
- o locate and erase within files.

10.20.5 Implementation Class A2: Full File Access

Implementation class A2 provides the function of transfer of and access to files with unstructured or flat constraint sets at the reliable file service level. This includes support of the document types:

- o FTAM-3 (unstructured binary files)
- o NBS-2 (unstructured text files)
- o NBS-3 (unstructured text files)
- o NBS-4 (sequential text files)
- o NBS-5 (sequential text files)
- o NBS-6 (sequential files)
- o NBS-7 (random access files)
- o NBS-8 (indexed sequential files)

This implementation class supports file transfer and file access, that is, the ability to:

- o read from a complete file, or from a series of FADUs which are identified by key or by position,
- o write (replace, extend, insert) to a file or an FADU,
- o locate and erase within files.

10.20.6 Implementation Class M1: Management

Implementation class M1 provides the function for an Initiator to manage the files within the Virtual Filestore, to which access is provided by the Responder. Management includes the services of:

- o creating a file
- o deleting a file
- o reading attributes of a file
- o changing attributes of a file.

10.21 HARMONIZATION

The implementation classes for File Transfer, File Access and Management correspond to the profiles of SPAG (Standards Promotion and Application Group) in Europe, so that interworking will be possible. Those profiles are described in the 'Guide to the Use of Standards' (GUS); they will also be the basis for the Functional Standards as defined by CEN/CENELEC (Comite Europeenne de Normalization).

Tbl. 10.6 Implementation classes (NBS) and profiles (SPAG)

Implementation Class	SPAG Profile
T1	A111
T2	A112
T3	A113
A1	A122
A2	A123
M1	A13

APPENDIX 10A: FTAM DOCUMENT TYPES

Part 1: Document Types
Part 2: Constraint Sets
Part 3: Abstract Syntaxes
Part 4: Transfer Syntaxes

Part 1: Document Types

Entry number: NBS-1

Document Type Name:

{ISO registration-authority NBS FTAM( ) document(6) UNDEF(0)}

Document Descriptor Value: unstructured binary file

Document Semantics:

The document consists of a single data unit. The data unit consists of an unbounded sequence of data elements. Each data element is an octet string.
Note: The boundaries between transferred data elements are not maintained by the filestore.

Scope and Field of Application:

This document type defines the contents of a file for storage and for transfer using FTAM.

Constraint Set Name:

{ISO standard 8571 constraint set name(5) unstructured(1)}

Abstract Syntax:

The abstract syntax of each Data Element is an instance of the ASN.1 data type OctetString.

Abstract Syntax Name:

{ISO registration-authority NBS FTAM( ) abstract syntax(2) NBS-AS1(0)}

Transfer Syntax:

An implementation supporting this abstract syntax shall support a transfer syntax for the data unit obtained by applying the ASN.1 Basic Encoding Rules (ISO 8825) to the data elements and concatenating the resulting octets for all data elements of the data unit.

Note: This transfer syntax is not self-delimiting with respect to the data unit, i.e., there is no ASN.1 encoding for the complete data unit.

Implementations may optionally support other named transfer syntaxes for this abstract syntax.

Transfer Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1(0)}

Concatenation:

Concatenation of this type with itself is possible, and produces a document of the same type consisting of one data unit which is the concatenation of the octet string(s) from one file with the octet string(s) of the other file.

Note: The boundary of the original octet string(s) is no longer visible.

Simplifications:

A document of this type cannot be accessed as any other document type.

Entry Number: NBS-2

Document Type Name:

{ISO registration-authority NBS FTAM( ) document(6) VARCRLF(1)}

Document Descriptor Value: unstructured text file

Document Semantics:

The document consists of a single data unit. The data unit consists of an unbounded sequence of data elements. Each data element is an IA5String. The last two characters of each data element are carriage return followed by line feed. Neither the character carriage return nor the character line feed may appear elsewhere in the data element.

Note: The boundaries between transferred data elements are not maintained by the filestore.

Scope and Field of Application:

The document type defines the contents of a file for transfer using FTAM.

Note that this document type should only be used for transferring entire text files in the case where NBS-4 is not supported. It has an implicit structure which allows, for example, text files stored in UNIX format (lines terminated by LF) to be converted to a format in which lines are terminated by CR followed by LF and vice versa.

Constraint Set Name:

{ISO standard 8571 constraint set name(5) unstructured(1)}

Abstract Syntax:

The abstract syntax of each Data Element is an instance of the ASN.1 data type IA5String.

Abstract Syntax Name:

{ISO registration-authority NBS FTAM( ) abstract syntax(2) NBS-AS1(0)}

Transfer Syntax:

An implementation supporting this abstract syntax shall support a transfer syntax for the data unit obtained by applying the ASN.1 Basic Encoding Rules (ISO 8825) to the data elements and concatenating the resulting octets for all data elements of the data unit.

Note: This transfer syntax is not self-delimiting, with respect to the data unit, i.e., there is no ASN.1 encoding for the complete data unit.

Implementations may optionally support other named transfer syntaxes for this abstract syntax.

Transfer Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1(0)}

Concatenation:

Concatenation of this type with itself is possible, and produces a document of the same type consisting of one data unit which is the concatenation of the IA5 String(s) from one file with the IA5 String(s) of the other file.

Note: The boundary of the original octet string(s) is no longer visible.

Simplification:

A document of this type can be accessed as a document of type NBS-1 (allowed only when reading the file) by specifying a document type of NBS-1 in the Contents Type parameter of the F-OPEN request, and limiting access context to US on F-READ.

Entry Number: NBS-3

Document Type Name:

{ISO registration-authority NBS FTAM( ) document(6) 8859VARCRLF(2)}

Document Descriptor Value: unstructured text file

Document Semantics:

The document consists of a single data unit. The data unit consists of an unbounded sequence of data elements. Each data element is an 8859String. The last two characters of each data element are carriage return followed by line feed. Neither the character carriage return nor the character line feed may appear elsewhere in the data element.

Note: The boundaries between transferred data elements are not maintained by the filestore.

Scope and Field of Application:

The document type defines the contents of a file for transfer using FTAM.

Note that this document type should only be used for transferring entire text files in the case where NBS-5 is not supported. It has an implicit structure which allows, for example, text files stored in UNIX format (lines terminated by LF) to be converted to a format in which lines are terminated by CR followed by LF and vice versa.

Constraint Set Name:

{ISO standard 8571 constraint set name(5) unstructured(1)}

Abstract Syntax:

The abstract syntax of each data element is an instance of the data type 8859String.

Abstract Syntax Name:

{ISO registration-authority NBS FTAM( ) abstract syntax(2) NBS-AS1(0)}

Transfer Syntax:

An implementation supporting this abstract syntax shall support a transfer syntax for the data unit obtained by applying the ASN.1 Basic Encoding Rules (ISO 8825) to the data elements and concatenating the resulting octets for all data elements of the data unit.

Note: This transfer syntax is not self-delimiting, with respect to the data unit, i.e., there is no ASN.1 encoding for the complete data unit.

Implementations may optionally support other named transfer syntaxes for this abstract syntax.

Transfer Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1 (0)}

Concatenation:

Concatenation of a document of this type with another document of this type is possible and produces a document of the same type consisting of one data unit which is the concatenation of the 8859 string(s) of one file with the 8859 string(s) of the other file.

Note: The boundary of the original OctetString is no longer visible.

Simplification:

A document of type NBS-3 can be accessed as a document of type NBS-1 (allowed only when reading the file) by specifying a document type of NBS-1 in the Contents Type parameter of the F-OPENrequest, and limiting access context to US on F-READ.

Entry number: NBS-4

Document type name:

{ISO registration-authority NBS FTAM( ) document(6) Text(3) parameter}

Note: "parameter" is a parameter which will be appended to the registered identifier in an OBJECT IDENTIFIER.

Document Descriptor Value: Sequential Text File

Document Semantics:

The document consists of an unbounded series of data units. Each data unit contains one data element which is a character string. Each character is taken from the IA5 character set.

Scope and Field of Application:

The document type defines the contents of a file for storage and for transfer using FTAM.

Note: Storage refers to apparent storage within the virtual filestore.

Constraint Set Name:

{ISO standard 8571 constraint set name(5) sequential flat(2)}

Additional Constraints:

FADU Identity will be limited to begin, end, first and next.

Abstract Syntax:

The abstract syntax of each data element is an instance of the data type IA5String.

The abstract syntax of each data unit is specified by the parameter in the Document Type Name, which determines the maximum length of the data unit.

Abstract Syntax Name:

{ISO registration-authority NBS FTAM( ) abstract syntax(2) NBS-AS1(0)}

Transfer Syntax:

An implementation supporting this abstract syntax shall support a transfer syntax for each data unit obtained by applying the ASN.1 Basic Encoding Rules (ISO 8825) to the data element and concatenating the resulting octets for the single data element of the data unit.

Note: This transfer syntax is self-delimiting in that there is a one-to-one correspondence between the data unit and its data element.

Implementations may optionally support other named transfer syntaxes for this abstract syntax.

Transfer Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1(0)}

Concatenation:

Concatenation of a document of this type with another document of this type is possible and produces a document of the same type consisting of a series of IA5Strings which is the result of placing the series of IA5Strings from one file of this type after the last IA5String in the original file.

Note: The boundary of the original sequence is no longer visible.

Simplification:

A document of type NBS-4 can be accessed as a document of type NBS-1 (allowed only when reading the file) by specifying a document type of NBS-1 in the Contents Type parameter on the F-OPENrequest, and limiting access context to US on F-READ.

Entry number: NBS-5

Document Type Name:

{ISO registration-authority NBS FTAM( ) document(6) 8859Text(4) parameter}

Note: "parameter" is a parameter which will be appended to the registered identifier in an Object Identifier.

Document Descriptor Value: Sequential Text File

Document Semantics:

The document consists of an unbounded series of data units. Each data unit contains one data element which is a character string. Each character is taken from the ISO 8859/1 character set.

Scope and Field of Application:

The document type defines the contents of a file for storage and transfer using FTAM.

Note: Storage refers to apparent storage within the virtual filestore.

Constraint Set Name:

{ISO standard 8571 constraint set name(5) sequential flat(2)}

Additional Constraints:

FADU Identity will be limited to begin, end, first and next.

Abstract Syntax:

The abstract syntax of each data element is an instance of the data type 8859String.

The abstract syntax of each data unit is specified by the parameter in the Document Type Name, which determines the maximum length of the data unit.

Abstract Syntax Name:

{ISO registration-authority NBS FTAM( ) abstract syntax(2) NBS-AS1(0)}

Transfer Syntax:

An implementation supporting this abstract syntax shall support a transfer syntax for each data unit obtained by applying the ASN.1 Basic Encoding Rules (ISO 8825) to the data element and concatenating the resulting octets for the single data element of the data unit.

Note: This transfer syntax is self-delimiting in that there is a one-to-one correspondence between the data unit and its data element.

Implementations may optionally support other named transfer syntaxes for this abstract syntax.

Transfer Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1 (0)}

Concatenation:

Concatenation of a document of this type with another document of this type is possible and produces a document of the same type consisting of a series of 8859Strings which is the result of placing the series of 8859Strings from one file immediately following the last 8859String in the original file.

Note: The boundary of the original series is no longer visible.

Simplification:

A document of type NBS-5 can be accessed as a document of type NBS-1 (allowed only when reading the file) by specifying a document type of NBS-1 in the Contents Type parameter on the F-OPENrequest, and limiting access context to US on F-READ.

Entry Number: NBS-6

Document Type Name:

{ISO registration-authority NBS FTAM( ) document(6) SEQUENTIAL(5) parameter}

Note: "parameter" is a parameter which will be appended to the registered identifier in an Object Identifier.

Document Descriptor Value: Sequential File

Document Semantics:

The document consists of an unbounded series of data units. Each data unit contains an unbounded series of data elements. Each data element is a data type from the set of primitive data types defined in the main body of this document. Each data unit contains the same data element types in the same order as all other data units.

Note: The series of data elements for a data unit in a specific file is bounded by the parameter.

Scope and Field of Application:

The document type defines the contents of a file for storage and transfer using FTAM.

Note: Storage refers to apparent storage within the virtual filestore.

Constraint Set Name:

{ISO standard 8571 constraint set name(5) sequential flat(2)}

Additional Constraints:

FADU Identity will be limited to begin, end, first, and next.

Abstract Syntax:

The abstract syntax of each data element is an instance of one of the primitive data types defined in the main body of this document.

The Abstract syntax of each data unit is specified by the parameter in the Document Type Name which determines types and number of data elements in the data unit.

Abstract Syntax Name:

{ISO registration-authority NBS FTAM( ) abstract syntax(2) NBS-AS1(0)}
optionally, {ISO registration-authority NBS FTAM( ) abstract syntax(2)

NBS-AS2(1)}

Transfer Syntax:

An implementation supporting this abstract syntax shall support a transfer syntax for each data unit obtained by applying the ASN.1 Basic Encoding Rules (ISO 8825) to its data elements and concatenating the resulting octets for all data elements of the data unit.

Note: This transfer syntax is not self delimiting with respect to the data units, i.e., there is no ASN.1 encoding for complete data units. However, data unit boundaries may be recognized, since types and number of data elements are the same for each data unit and are determined by the parameter in the Document Type Name.

Implementations may optionally support other named transfer syntaxes for this abstract syntax.

Transfer Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1(0)}

Concatenation:

Concatenation of a document of this type with another document of this type is possible and produces a document of the same type consisting of a series of data units which is the result of placing the series of data units from one file immediately following the last data unit of the original file.

Note: The boundary of the original file is no longer visible.

Simplification:

A document of type NBS-6 can be accessed as a document of type NBS-1 (allowed only when reading the file) by specifying a document type of NBS-1 in the Contents Type parameter in the F-OPENrequest, and limiting access context to US on F-READ.

Entry number: NBS-7

Document Type Name:

{ISO registration-authority NBS FTAM( ) document(6) RANDOM(6) parameter}

Note: "parameter" is a parameter which will be appended to the registered identifier in an Object Identifier.

Document Descriptor Value: Random Access File

Document Semantics:

The document consists of an unbounded series of data units. Each data unit contains an unbounded series of data elements. Each data element is a data type from the set of primitive data types defined in the main body of this document. Each data unit contains the same data types in the same order as all other data units in the file.

Note: The series of data elements for a data unit in a specific file is bounded by the parameter.

Scope and Field of Application:

The document type defines the contents of a file for storage and transfer using FTAM.

Note: Storage refers to apparent storage within the virtual filestore.

Constraint Set Name:

{ISO registration-authority NBS FTAM( ) constraint set name(5)
NBS Ordered Flat(2)}

Abstract Syntax:

The abstract syntax of each data element is an instance of one of the primitive data types defined in the main body of this document.

The abstract syntax of each data unit is specified by the parameter in the Document Type Name which determines types and number of data elements in the data unit.

Abstract Syntax Name:

{ISO registration-authority NBS FTAM( ) abstract syntax(2) NBS-AS1(0)} optionally, {ISO registration-authority NBS FTAM( ) abstract syntax(2) NBS-AS2(1)}

Transfer Syntax:

An implementation supporting this abstract syntax shall support a transfer syntax for each data unit obtained by applying the ASN.1 Basic Encoding Rules (ISO 8825) to its data elements and concatenating the resulting octets for all data elements of the data unit.

Note: This transfer syntax is not self-delimiting with respect to the data units, i.e., there is no ASN.1 encoding for complete data units. However, data unit boundaries may be recognized, since types and number of data elements are the same for each data unit and are determined by the parameter in the Document Type Name.

Implementations may optionally support other named transfer syntaxes for this abstract syntax.

Transfer Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1(0)}

Concatenation:

Concatenation of a document of this type with another document of this type is possible and produces a document of the same type consisting of a series of data units which is the result of placing the series of data units from one file immediately following the last data unit of the original file.

Note: The boundary of the original file is no longer visible.

Simplification:

A document of type NBS-7 can be accessed as a document of type NBS-1 (allowed only when reading the file) by specifying a document type of NBS-1 in the Contents Type parameter of the F-OPENrequest, and limiting access context to US on F-READ.

A document of type NBS-7 can be accessed as a document of type NBS-6 (allowed only when reading the file) by specifying a document type of NBS-6 in the Contents Type parameter of the F-OPENrequest.

Entry Number: NBS-8

Document Type Name:

{ISO registration-authority NBS FTAM( ) document(6) INDEXED(7) p1 p2 p3}

Note: "p1","p2" and "p3" are parameters which will be appended to the registered identifer in an Object Identifier.

Document Descriptor Value: Indexed Sequential File

Document Semantics:

The document consists of an unbounded series of data units. Each data unit is an unbounded series of data elements. Each data element is a data type from the set of primitive data types defined in the main body of this document. Each data unit contains the same data types in the same order as all other data units in the file.

Each data unit in the file has a key associated with it. The key of each data unit is of the same data type as the key of all other data units in the file and is a single data element from the set of primitive data types defined in the main body of this document.

The primitive data types and minimum size range of each unit which an implementation must accept as a key value are given in the following table.

Key Type	Minimum Range (octets)
ASN.1 Integer	(1-2)
ANS.1 IA5String	(0-16)
NBS-AS1 8859String	(0-16)
ASN.1 OctetString	(0-16)
ASN.1 GeneralizedTime	
ASN.1 UniversalTime	
NBS-AS2 FloatingPoint	

Note: The series of data elements for a data unit in a specific file is bounded by the parameter.

Scope and Field of Application:

The document type defines the contents of a file for storage and for transfer using FTAM.

Constraint Set Name:

{ISO registration-authority NBS FTAM( ) constraint set name(2)
Indexed Flat(1)}

Abstract Syntax:

The abstract syntax of each data element is an instance of one of the primitive data types defined in the main body of this document.

The Abstract syntax of each data unit is specified by the parameter p1 in the Document Type Name which determines types and number of data elements in the data unit.

The Abstract syntax of the data unit key (FADU Identifier) is specified by the parameter p2.

The position of the key in the data unit is specified by the parameter p3.

Note: p3 = 0 implies the key is not part of the data
p3 > 0 specifies actual data element in the data unit.

Abstract Syntax Name:

The first data element in data unit is numbered 1.

{ISO registration-authority NBS FTAM( ) abstract syntax(2) NBS-AS1(0)}
optionally, {ISO registration-authority NBS FTAM( ) abstract syntax(2)
NBS-AS2(1)}

Transfer Syntax:

An implementation supporting this abstract syntax shall support a transfer syntax for each data unit obtained by applying the ASN.1 Basic Encoding Rules (ISO 8825) to its data elements and concatenating the resulting octets for all data elements in the data unit.

Note: This transfer syntax is not self-delimiting with respect to the data units, i.e., there is no ASN.1 encoding for complete data units. However, data unit boundaries may be recognized, since types and number of data elements are the same for each data unit and are determined by parameter "p1" in the Document Type Name.

Implementations may optionally support other named transfer syntaxes for this abstract syntax.

Transfer Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1(0)}

Concatenation:

A document of this type may not be concatenated with a document of this type or any other type.

Simplification:

A document of type NBS-8 can be accessed as a document of type NBS-1 (allowed only when reading the file) by specifying document type NBS-1 in the Contents Type parameter on the F-OPENrequest, and limiting access context to US on F-READ.

A document of type NBS-8 can be accessed as a document of type NBS-6 (allowed only when reading the file) by specifying document type NBS-6 in the Contents Type parameter on the F-OPENrequest.

Entry Number: NBS-9

Document Type Name:

{ISO registration-authority NBS FTAM( ) document(6) FILE_DIRECTORY(8)}

Document Descriptor Value: FileDirectory File

Document Semantics:

The document consists of an unbounded sequence of data units. Each data unit consists of one and only one data element of type FileDirectoryEntry (a complex data type defined in this appendix).

Scope and Field of Application:

This document defines the contents of a file for transfer (not for storage) using FTAM.

Constraint Set Name:

{ISO standard 8571 constraint set name(5) sequential flat (2)}

Additional Constraints:

FileDirectory Files may be Selected, Opened, Read, Closed, Created, and Deleted. They may not be Written or Modified (except as a side-effect of actions performed on individual files contained within a FileDirectory). DataUnits within a FileDirectory may only be accessed sequentially.

Abstract Syntax:

An indefinite series of data units. Each data unit contains one data element of type FileDirectoryEntry. Each data element consists of a required FileName and ContentsTypeName attributes.

Abstract Syntax Name:

{ISO registration-authority NBS FTAM( ) abstract syntax(3) filedirectory entry(0)}

Transfer Syntax:

An implementation supporting this abstract syntax shall support a transfer syntax for each data unit obtained by applying ASN.1 Basic Encoding Rules (ISO 8825) to its data element and concatenating the resulting octets for the single data element of the data unit.

Note: This transfer syntax is self-delimiting in that there is a one-to -one correspondence between the data unit and its data element.

Implementations may also support other named transfer syntaxes for this

abstract syntax.

Transfer Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1(0)}

Concatenation:

A document of this type cannot be concatenated with a document of this type or any other type.

Simplification:

A document of this type cannot be simplified.

Part 2: Constraint Sets

Constraint Set Title: NBS-Ordered Flat

Constraint Set Name:

{ISO registration-authority NBS FTAM( ) constraint set name(5)
NBS Ordered Flat(2)}

Field of Application: Files which are structured into a sequence of individual FADUs and to which access may be made on a FADU basis by position in the sequence.

Node Names: none

Actions: Locate, Read, Replace, Insert, Erase

Special Action Parameters: none

Special Action Semantics:
Erase: Used on the root node to empty the file. When used on a leaf node, it leaves a FADU with no associated data unit.
Insert: Allowed only at end of file. The new node is inserted following all existing nodes in the file or on a leaf node with no existing data unit. The inserted data unit is associated with the currently existing leaf node.

Available Access Contexts: HA, FA, UA, US

Erase and Locate Context: HA

Constraints on Structure: The root node shall not have an associated data unit. All children of the root node shall be leaf nodes and may have an associated data unit. All arcs from the root node shall be of length one.

Creation State: Root node without an associated data unit.

FADU Identity: begin, end, first, last,
current, next, previous,
traversal number (greater than or equal to one)

Location After Open: root node

Beginning of File: root node

End of File: No node is selected. Previous gives the last node in the traversal sequence, current and next result in an error.

Constraint Set Title: Indexed Flat

Constraint Set Name:

{ISO registration-authority NBS FTAM( ) constraint set name(5)
NBS Indexed Flat(1)}

Field of Application: This constraint set is for representing single key ISAM files where the keys are the Node Names for the leaf nodes. The keys are restricted to being single primitive data types, and restricted to all keys being of the same primitive data type.

Node Names: Any single primitive data type.

Actions: Locate, Read, Replace, Insert, Erase

Special Action Parameters: none

Special Action Semantics:

Locate: The specified FADU is made the current FADU. If the Node Name form is used, the most recently inserted FADU with the specified FADU at level 1 is located.

Insert: Insert the specified FADU (level 1 only) in the lexical order of the key primitive data types. If there is already another FADU with the specified Node Name, insert the new one after (in pre-order traversal) the existing FADUs and indicate that this was done via a diagnostic on TRANSFER_END.

Replace: Allowed only at leaves and only in access context US (DU only w/o delimiters).

Erase: If the addressed FADU is the root, the file is reduced to the initial state.

Available Access Contexts: HA, FA, UA, US

Erase and Locate Context: HA

Constraints on Structure: The root node shall not have an associated data unit and/or Node Name. All children of the root node shall be leaf nodes and shall have an associated data unit and Node Name. All arcs from the root node shall be of length one. Some primitive types may not be supported as keys.

Creation:	Root node without an associated data unit or Node Name.
FADU Identity:	begin, end, current, next, previous, Node-Name, Sequence of Node-Names
Location After Open:	root node
Beginning of File:	root node
End of File:	No node is selected. Previous gives the last node in the traversal sequence, current and next result in an error.

Abstract Syntax: NBS-AS1

Abstract Syntax Name:

{ISO registration-authority NBS FTAM( ) Abstract Syntax(2) Basic(0)}

Abstract Syntax Definition:

```
DE ::=Choice{INTEGER,
             BITSTRING,
             BOOLEAN,
             IA5String,
             8859String,
             OCTETSTRING,
             UniversalTime,
             GeneralizedTime,
             Null}
8859String ::=[PRIVATE 1] Implicit 8859CharacterString

Transfer syntax name:     -- 8859CharacterString is a string of characters from
                             the ISO 3859 character set
{ISO registration-authority NBS FTAM( ) Transfer Syntax(4) NBS-TS1 (0)}
```

Abstract Syntax: NBS-AS2

Abstract Syntax Name:

{ISO registration-authority NBS FTAM( ) abstract syntax(2) FloatingPoint(1)}

Abstract Syntax Definition:

```
      FloatingPointNumber ::= [PRIVATE 0] CHOICE
              {
                 finite [0] IMPLICIT SEQUENCE
                 {
                          Sign,
                          mantissa BITSTRING,
                          exponent INTEGER
                       },
                  infinity [1] IMPLICIT Sign,
                  signalling NaN [2] Implicit NaN,
                  quietNaN [3] IMPLICIT NaN,
                  zero [4] IMPLICIT NULL
              }
       Sign ::= INTEGER {positive(0), negative(1)}

       NaN ::= INTEGER
```

Transfer Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1(0)}

Abstract Syntax: NBS-AS3

Abstract Syntax Name:

{ISO registration-authority NBS FTAM( ) abstract syntax(2) FileDirectoryElement(2)}

Abstract Syntax Definition:

```
    FileDirectoryEntry ::=[PRIVATE 2] IMPLICIT SEQUENCE{
                                        fileName GraphicString,
                                          ContentsType}
  ContentsType ::=CHOICE {
                  document-type-Name[0] IMPLICIT OBJECT IDENTIFIER,
                  constraint-set-and-abstract-syntax [1] IMPLICIT SEQUENCE{
                           constraint-set-Name[0] IMPLICIT OBJECT IDENTIFIER,
                           Abstract-syntax-Name[1] IMPLICIT OBJECT IDENTIFIER}}
```

Transfer Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1(0)}

Part 4: Transfer Syntaxes

Transfer Syntax: NBS-TS1

Transfer Syntax Name:

{ISO registration-authority NBS FTAM( ) transfer syntax(3) NBS-TS1(0)}

Encoding Rules:

ASN.1 Basic Encoding Rules shall apply.

The first bit of a "mantissa" must be "1".

Transfer Syntax Definition:

The transfer syntax shall be that which results from applying the encoding rules described above to the individual data elements.

APPENDIX 10B: KNOWN ERRORS IN ISO AND CCITT DOCUMENTS

This appendix lists errors that are known in ISO and CCITT documents. Known errors are removed from this appendix when corrected text is available from ISO and CCITT. This appendix is for information only.

FTAM DIS Errors

Following is a list of known errors in the FTAM DIS which are expected to be corrected in the forthcoming IS. These errors are listed here for information only and will be removed when the IS text is available.

1. The Access Context parameter on the F-READrequest and F-WRITErequest is specified as OPTIONAL instead of mandatory in the FTAM protocol abstract syntax.

The expected correction is to make this parameter mandatory.

2. The Contents Type List is defined as a SEQUENCE of Document Type Name and Constraint-Set, Abstract-Syntax in the FTAM protocol abstract syntax.

The expected solution is to use the following definition:

```
Contents-Type-List ::= IMPLICIT SEQUENCE OF CHOICE{
   document-types [0] IMPLICIT Document-Type-Name,
   constraint-set-and-abstract-syntax[1] IMPLICIT SEQUENCE{
      constraint-sets [0] IMPLICIT Constraint-Set-Name,
      abstract-syntax [1] IMPLICIT Abstract-Syntax-Name}}
```

3. The processing-mode parameter on the F-OPEN request is not defined correctly in the FTAM protocol abstract syntax.

The expected correction is:

```
processing-mode [0] IMPLICIT BITSTRING {
                    read (0),
                    insert (1),
                    replace (2),
                    erase (3),
                    extend (4) }
```

4. In Part 2 Clause 5.3.2 the application tags conflict with tags in Part 4 in the protocol abstract syntax.

The expected correction is to use the following definitions in Part 2 Clause 5.3.2:

```
Node-Descriptor-Data-Element ::= [Application 21] IMPLICIT...
Enter-Subtree-Data-Element   ::= [Application 22] Implicit Null...
Exit-Subtree-Data-Element    ::= [Application 23] IMPLICIT Null...
```

5. An object identifier for the CASE protocol abstract syntax is not defined.

The expected correction is:

{iso standard 8650 abstract syntax (1) acse (1)}

6. In 8571/2, Clause 5.3.5 (f) contradicts Clause 7.1 (i). Clause 7.1 (i) is correct.

11. PHASE 3 FTAM IMPLEMENTATION SPECIFICATION

11.1 INTRODUCTION

Will be included in Phase 3:

- o specify error control procedures
- o specify Recovery and Restart Data Transfer functional units
- o specify concurrency control parameters
- o specify the use of directory services
- o specify implementation of character set ISO 6937

May be included in Phase 3:

- o new document types/constraint sets
- o define the use of Access Control
- o specify the use of Presentation Context Management functional unit
- o specify implementation of Filestore Management
- o define filename convention
- o specify overlapped access

11.2 SCOPE AND FIELD OF APPLICATION

Phase 3 FTAM implicitly includes all of the implementation agreements and conformance requirements of Phase 2 FTAM. The Phase 3 FTAM specification will only specify extensions to Phase 2 FTAM and implementations of Phase 3 FTAM are expected to also implement Phase 2 FTAM.

Phase 3 FTAM will include at least the following:

- o Specification of Error Control Procedures
- o Specification of Recovery and Restart Data Transfer functional units
- o Specification of concurrency control parameters
- o Use of character set ISO 6937
- o Specification of the use of directory services

11.3 STATUS

The Phase 3 FTAM specification has been accepted as the basis for further work on this specification. It has been given only preliminary review by the FTAM SIG. It is subject to much change in all sections at future meetings. This specification is expected to be completed by December 1988.

11.4 ERRATA

11.5 ASSUMPTIONS

1. Implementations will be based on the ISO 8571 DIS version of FTAM. When the IS text is approved following the close of the DIS ballot the agreements will be modified as necessary to meet the IS specifications.

2. The following documents are required for reference:

   - o ISO 8571 Parts 1-4 FTAM
   - o ISO 8649, ISO 8650 ACSE
   - o ISO 8822, ISO 8823 Presentation
   - o ISO 8326, ISO 8327 Session
   - o ISO 6937 character set
   - o ISO 8571 PDAD1 - Filestore Management

11.6 FILESTORE AGREEMENTS

11.7 SERVICE AGREEMENTS

11.7.1 FTAM Service Level Agreements

Implementation of both the User Correctable File Service (UCFS) and the Reliable File Service (RFS) is defined. Implementation of the UCFS implies the ability to negotiate for the use of the Recovery and/or the Restart Data Transfer functional units. Implementation of the RFS implies implementation of the File Error Recovery Protocol Machine (FERPM) as specified in Annex A of the Part 4 FTAM standard.

11.7.2 Service Class Agreements

Implementation of a service class is defined as the ability to negotiate for the use of the service class in any service level negotiated, and the ability to provide the services of the service class in an ISO conformant way.

There are no service classes specified in addition to those specified in Section 11.2.4.

11.7.3 Functional Unit Agreements

Implementation of a functional unit is defined as the ability to negotiate for the use of the functional unit in any service level and service class where the use of that functional unit is allowed, and the ability to provide the services of the functional unit in an ISO conformant way.

Implementation of the following functional units is defined.

- o Recovery
- o Restart Data Transfer

11.7.4 Error Recovery

1.When a class I, II or III error occurs, the docket will always be present as long as the association is not terminated. Recovery from a class I, II or III error is defined as long as the association is not terminated. Once the association is terminated, recovery from a class I, II or III error is not possible.

2. When a class IV error occurs, the length of time the docket is maintained is determined by the local system. Recovery from a class IV error is only possible as long as both end systems maintain the docket.

3. If the RFS has been selected, the number of times the error recovery entity will try to recover from an error is an implementation option.

4. If an error occurs when the RFS has been selected, the error recovery entity will wait at least the amount of time specified by the "suggested delay" field of the diagnostic parameter before attempting to recover. If "suggested delay" time is not specified, the default delay time is an implementation option.

11.7.5 Concurrency

Concurrency controls required for the access to a file may be specified by the initiating service user.

If, when the request (select, create or open) is received by the filestore there is currently no other user of the file, the concurrency controls requested will be applied (local filestore considerations may limit the controls which may be accepted).

If, when the request (select, create or open) is received by the filestore there is currently at least one other user of the file, the concurrency controls requested will be checked for violation of the concurrency controls currently applied to the file by each current file accessor for each action. If the controls requested do not violate the controls currently in place, the controls requested will be applied to the regime being established.

The following table defines compatibility of requested concurrency controls with controls already in use.

Tbl. 11.1 Concurrency negotiation rules

control request / control currently applied	not required	shared	exclusive	no access
not required	A	A	A	A
shared	A	A	N	N
exclusive	A	N	N	N
no access	A	N	N	N

Key: A - accepted
N - send negative response

Note: Concurrency controls must be maintained individually for each file accessor.

If the concurrency controls requested by the initiating service user are not acceptable to the responding service user due to local filestore considerations, or due to violation of concurrency controls already imposed by other users, the request must be rejected.

If no requested concurrency controls are specified by the initiating service user the following defaults will be applied.

action	control
read	shared
insert	exclusive
replace	exclusive
extend	exclusive
read attrib	shared
change attrib	exclusive
delete file	exclusive

Note: Use of the concurrency control parameter requires the successful negotiation of the availability of storage group of file attributes.

11.8 PROTOCOL AGREEMENTS

11.9 CONFORMANCE

In addition to the specific conformance requirements specified in the following subsections, conformance to this specification requires:

- o conformance to ISO 8571
- o conformance to Phase 2 FTAM specified in Section 10.2.

11.9.1 Initiators

Every implementations of an FTAM initiator shall support:

1. the Recovery protocol and its mandatory parameters with minimum ranges,
2. the Restart Data Transfer protocol and its mandatory parameters with minimum ranges,
3. the use of the concurrency control parameter on the F-SELECT, F-CREATE, and F-OPEN,

and support the applicable procedures, defined in ISO 8571/4 Clauses 8.13, 15.3, 16.3 and 17.3. To support the above protocols and procedures the implementation shall be able to:

1. request the use of either the UCFS or the RFS,
2. request the use of either Recovery functional unit, Restart Data Transfer functional unit of both when UCFS has been selected,
3. request the storage group of file attributes with the "attribute groups" parameter.

11.9.2 Responders

Every implementation of an FTAM responder shall support:

1. the Recovery protocol and its mandatory parameters with minimum ranges,
2. the Restart Data Transfer protocol and its mandatory parameters with minimum ranges,
3. the use of the concurrency control parameter on the F-SELECT, F-CREATE, and F-OPEN,

and support the applicable procedures, defined in ISO 8571/4 Clauses 9.13, 15.3, 16.3 and 17.3. To support the above protocols and procedures the implementation shall be able to:

1. accept requests for the use of either the UCFS or the RFS,
2. accept requests for the use of either the Recovery functional unit, Restart Data Transfer functional unit or both when UCFS has been selected,
3. accept requests for the storage group of file attributes with the "attribute groups" parameter.

11.9.3 Error Recovery Procedures

Every implementation of either an initiator or responder shall support the File Error Recovery Protocol Machine and support the procedures, defined in ISO 8571/4 Clauses 18 and 19. To support the above procedures an implementation shall:

1. maintain the docket after a class IV error for at least 1 hour after the occurrence of the error,
2. when the RFS has been selected, attempt recovery from each error at least once,
3. when the RFS has been selected, support a default delay time before attempting recovery of 1 minute.

13. CCITT 1984 X.400 BASED MESSAGE HANDLING SYSTEM

13.1 INTRODUCTION

This is an implementation agreement developed by the Implementor's Workshop sponsored by the U.S. National Bureau of Standards to promote the useful exchange of data between devices manufactured by different vendors. This agreement is based on, and employs protocols developed in accord with, the OSI Reference Model. While this agreement introduces no new protocols, it eliminates ambiguities in interpretations.

This is an implementation agreement for a Message Handling System (MHS) based on the X.400-series of Recommendations (1984) and Version 5 of the X.400 Series Implementor's Guide from the CCITT. It is recommended that product vendors consult later versions of this guide. Figure 13-1 displays the layered structure of this agreement.

This agreement can be used over any Transport protocol class. In particular, this MHS agreement can be used over the Transport protocol class 0 used over CCITT X.25, described in section 7.6 of this document. In addition, this MHS agreement can be used over the Transport profiles used in support of MAP (Manufacturing Automation Protocol) or TOP (Technical and Office Protocols). Note that the MAP or TOP environment must support the reduced Basic Activity Subset (BAS) as defined in X.410.

The UAs and MTAs require access to directory and routing services. A Directory Service is to be provided for each (vendor-specific) domain. Except insofar as they must be capable of providing addressing and routing described hereunder, these services and associated protocols are not described by this agreement.

User Agent Layer	CCITT X.420
Message Transfer Agent Layer	CCITT X.411
Reliable Transfer Service Layer	CCITT X.410
Presentation Layer	CCITT X.410 sec. 4.2
Session Layer	CCITT X.225

Fig. 13-1 The layered structure of this implementation agreement

13.2 SCOPE

This agreement applies to Private Management Domains (PRMDs) and Administration Management Domains (ADMDs). Four boundary interfaces are specified:

(A) PRMD to PRMD;
(B) PRMD to ADMD;
(C) ADMD to ADMD.
(D) MTA to MTA (within a PRMD, e.g., for MTAs from different vendors.)

In case A, the PRMDs do not make use of MHS services provided by an ADMD. In cases B and C, UAs associated with an ADMD can be the source or destination for messages. Furthermore, in cases A and B, a PRMD can serve as a relay between MDs, and in cases B and C an ADMD can serve as a relay between MDs. Figure 13-2 illustrates the interfaces to which the agreement applies.

X.400 protocols other than the Message Transfer Protocol (P1) and the Interpersonal Messaging Protocol (P2) are beyond the scope of this agreement. Issues arising from the use of other protocols or relating to P1 components in support of other protocols are outside the scope of this document. This agreement describes the minimum level of services provided at each interface shown in Figure 13-2. Provision for the use

of the remaining services defined in the X.400 Series of Recommendations is outside the scope of this document.

With the exception of intra domain connections, this agreement does not cover message exchange between communicating entities within a domain even if these entities communicate via P1 or P2. Bilateral agreements between domains may be implemented in addition to the requirements stated in this document. Conformance to this agreement requires the ability to exchange messages without use of bilateral agreements.

PRMD = Private Management Domain

ADMD = Administration Management Domain

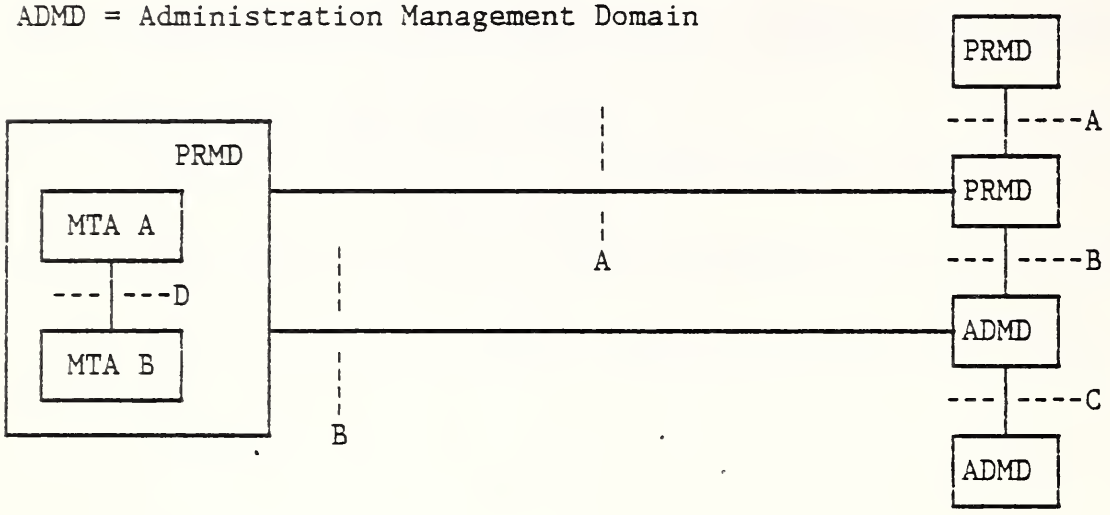

Fig. 13-2 This agreement applies to the interface between: (A) PRMD and PRMD; (B) PRMD and ADMD; (C) ADMD and ADMD; and (D) MTA and MTA

13.3 STATUS

This version of the X.400 based Message Handling System implementation agreements was completed on December 12, 1986. No further enhancements will be made to this version. See the next section--Errata.

13.4 ERRATA

This section shall contain any and all corrections and clarifications to this version of the agreements, that are identified after December 12, 1986. Each change shall be dated. Only text for clarification and

correction of errors shall appear here. The correction of typographical errors that do not affect the meaning of the text will not be noted.

13.5 PRMD to PRMD

13.5.1 Introduction

This section is limited in scope to issues arising from the direct connection (interface A in Figure 13-2) of two PRMDs. "Direct" means that no ADMD or relaying PRMD provides MHS services to facilitate message interchange. "Direct" does not exclude those instances for which Administrations happen to be ADMDs but are not providing X.400 services, that is, they are used only to provide lower layer services such as X.25. Figure 13-3 schematically represents the scope of this section.

These issues relate to the use of the UAL (User Agent Layer) and MTL (Message Transfer Layer) services, protocol elements, recommended practices and constraints. In particular, this section addresses the P1 and P2 protocols and their related services in a direct connection environment. This section describes the minimum level of services provided by a PRMD. Provision for the use of the remaining services defined in the X.400 Series of Recommendations is beyond the scope of this section.

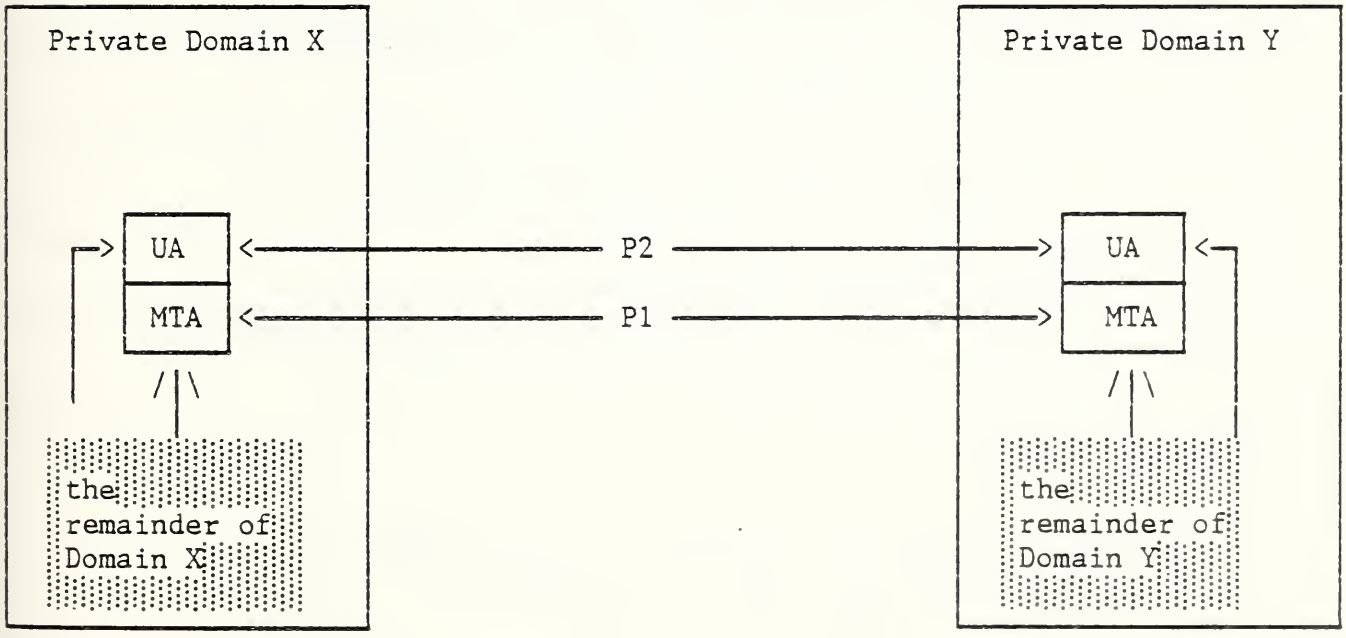

Fig. 13-3 Interconnection of private domains

13.5.2 Service Elements and Optional User Facilities

This section identifies those service elements and optional user facilities that must be provided in support of P1 and P2.

13.5.2.1 Classification of Support for Services

The classification of UA and MT-Service elements is used to define characteristics of equipment. Equipment can claim SUPPORT or NON-SUPPORT of a Service; in the case of UA-service elements, a separate classification is given for Origination and Reception.

The service provider is defined as the entity providing the service, in this case, the MTL or the UAL. The service user is either the MHS user or the UAL. The classification of provider and user relates to the sublayer for which the service element is defined.

13.5.2.1.1 Support (S)

a) Support means:

- The service provider makes the service element available to the service user.
- The service user gives adequate support to the MHS to invoke the service element or makes information associated with the service element available.

b) Support for Origination means that:

- The service provider makes the service element available to the service user for invocation.
- The service user gives adequate support to the end user of the MHS to invoke the service element.

c) Support for Reception means that:

- The service provider makes information associated with the service element available to the service user.

Note: A UA- or MT-service element can carry information from originator to recipient only if:

- the service element is available to the originator,
- the service element is available to the recipient, and
- all intermediate steps carry the information.

13.5.2.1.2 Non Support (N)

This means that the service provider is not required to make the service element available to the service user. However, the service provider should not regard the occurrence of the corresponding protocol elements as an error and should be able to relay such elements. Implementations making a profile available should indicate deviations (additions or deletions) with respect to the requirement in the profile.

13.5.2.1.3 Not Used (N/U)

This means that although the Recommendation allows this service element, this profile does not use it.

13.5.2.1.4 Not Applicable (N/A)

This means that this service element does not apply in this particular case (for originator or recipient).

13.5.2.2 Summary of Supported Services

a) Within a PRMD, a User Agent must support all P2 BASIC IPM Services (X.400) and all P2 ESSENTIAL IPM Optional user facilities (X.401) subject to the qualifiers listed in Appendix 13A.

b) Within a PRMD, a MTA must support all BASIC MT Services (X.400) and all ESSENTIAL MT optional user facilities (X.401) subject to the qualifiers listed in Appendix 13A.

c) No support is required of the additional optional user facilities of X.401.

13.5.2.3 MT Service Elements and Optional User Facilities

Tables 13-4 through 13-6 show the message transfer (MT) service elements and optional user facilities.

Tbl. 13-4 Basic MT service elements

Service Elements	Support (S) or Non-support (N)
Access Management	N/U[1]
Content Type Indication	S
Converted Indication	S
Delivery Time Stamp Indication	S
Message Identification	S
Non-delivery Notification	S
Original Encoded Information Types Indication	S
Registered Encoded Information Types	N/U[1]
Submission Time Stamp Indication	S

1 Not applicable to co-resident UA and MTA.

Tbl. 13-5 MT optional user facilities provided to the UA-selectable on a per-message basis

MT Optional User Facilities	Categorization	Support (S) or Non-support (N)
Alternate Recipient Allowed	E	S
Conversion Prohibition	E	S
Deferred Delivery	E	N[2]
Deferred Delivery Cancellation	E	N[2]
Delivery Notification	E	S
Disclosure of Other Recipients	E	N[3]
Explicit Conversion	A	N
Grade of Delivery Selection	E	S
Multi-destination Delivery	E	S
Prevention of Non-delivery Notification	A	N
Probe	E	N[4]
Return of Contents	A	N

Tbl. 13-6 MT optional user facilities provided to the UA agreed for any contractual period of time

MT Optional User Facilities	Categorization	Support (S) or Non-Support (N)
Alternate Recipient Assignment	A	N
Hold for Delivery	A	N/U
Implicit Conversion	A	N

E: Essential optional user facility.
A: Additional optional user facility.
2 A local facility subject to qualifiers in Appendix 13A.
3 Support not required for an originating MT user; support must be provided for recipient MT users.
4 Subject to qualifiers in Appendix 13A.

13:5.2.4 IPM Service Elements and Optional User Facilities

Tables 13-7 through 13-9 show the IPM service elements and optional user facilities.

Tbl. 13-7 Basic IPM service elements

Service Elements	Origination by UAs	Reception by UAs
Access Management	N/U[5]	N/U[5]
Content Type Indication	S	S
Converted Indication	N/A	S
Delivery Time Stamp Indication	N/A	S
Message Identification	S	S
Non-delivery Notification	S	N/A
Original Encoded Information Types Indication	S	S
Registered Encoded Information Types	N/A	N/A[5]
Submission Time Stamp Indication	S	S
IP-message Identification	S	S
Typed Body	S	S

5 Does not apply to co-resident UA and MTA.

Tbl. 13-8 IPM optional facilities agreed for a contractual period of time

Service Elements	Categorization	Support (S) or Non-Support (N)
Alternate Recipient Assignment	A	N
Hold for Delivery	A	N
Implicit Conversion	A	N

Tbl. 13-9 IPM optional user facilities selectable on a per-message basis

IPM Optional User Facilities	Origination by UAs	Reception by UAs
Alternate Recipient Allowed	A (N)	A (N)
Authorizing Users Indication	A (N)	E (S)
Auto-forwarded Indication	A (N)	E (S)
Blind Copy Recipient Indication	A (N)	E (S)
Body Part Encryption Indication	A (N)	E (S)
Conversion Prohibition	E (S)	E (S)
Cross-referencing Indication	A (N)	E (S)
Deferred Delivery	E (N)[6]	N/A
Deferred Delivery Cancellation	A (N/U)[6]	N/A
Delivery Notification	E (S)	N/A
Disclosure of Other Recipients	A (N)	E (S)
Expiry Date Indication	A (N)	E (S)
Explicit Conversion	A (N)	N/A
Forwarded IP-message Indication	A (N)	E (S)
Grade of Delivery Selection	E (S)	E (S)
Importance Indication	A (N)	E (S)
Multi-destination Delivery	E (S)	N/A
Multi-part Body	A (N)	E (S)
Non-receipt Notification	A (N)	A (N)
Obsoleting Indication	A (N)	E (S)
Originator Indication	E (S)	E (S)
Prevention of Non-delivery Notification	A (N)	N/A
Primary and Copy Recipients Indication	E (S)	E (S)
Probe	A (N)	N/A
Receipt Notification	A (N)	A (N)
Reply Request Indication	A (N)	E (S)
Replying IP-message Indication	E (S)	E (S)
Return of Contents	A (N)	N/A
Sensitivity Indication	A (N)	E (S)
Subject Indication	E (S)	E (S)

6 A local facility subject to qualifiers in Appendix 13A.

13.5.3 X.400 Protocol Definitions

This section reflects the agreements of the NBS/OSI Workshop regarding P1 and P2 protocol elements.

13.5.3.1 Protocol Classification

The protocol classifications are defined below in table 13-10:

1) UNSUPPORTED = X
These elements may be generated, but no specific processing should be expected in a relaying or delivering domain. A relaying domain must at least relay the semantics of the element. The absence of these elements should not be assumed, in a relaying or delivering domain, to convey any significance.

2) SUPPORTED = H
These elements may be generated. However, implementations are not required to be able to generate these elements. Appropriate actions shall be taken in a relaying or delivering domain.

3) GENERATABLE = G
Implementations must be able to generate and handle these protocol elements, although they are not necessarily present in all messages generated by implementations of this profile. Appropriate actions shall be taken in a relaying or delivering domain.

4) REQUIRED = R
Implementations of this profile must always generate this protocol element. However, its absence cannot be regarded as a protocol violation as other MHS implementations may not require this protocol element. Appropriate actions shall be taken in a relaying or delivering domain.

5) MANDATORY = M
This must occur in each message as per X.411 or X.420 as appropriate; absence is a protocol violation. Appropriate actions shall be taken in a relaying or delivering domain.

Tbl. 13-10 Protocol Classifications

13.5.3.2 General Statements on Pragmatic Constraints

a) Where a protocol element is defined as a choice of Numeric String and Printable String (i.e., Administration Domain Name and Private Domain Identifier), then a numeric value encoded as a printable string is equivalent to the same value encoded as a numeric string. This does not apply to the Country Name protocol element.

b) The maximum number of recipients in a single MPDU is 32K - 1 (that is, 32767). However, no individual limits on the number of occurrences (recipients) are placed on the following protocol elements: Authorizing Users, Primary Recipients, Copy Recipients, Blind Copy Recipients, Obsoletes and Cross References. Additionally, there is no limit on the number of Reply to Users. This is a local matter for the originating system.

c) Use of strings. A Printable String is defined in terms of the number of characters, which is the same number of octets. For T.61 strings the number of octets is twice the number of characters specified.

d) The ability to generate maximum size elements is not required, with the exception of the component fields in the Standard Attribute List, in which case it is required.

13.5.3.3 MPDU Size

The following agreements govern the size of MPDUs:

- All MTAEs must support at least one MPDU of at least one megabyte.

- The size of the largest MPDU supported by a UAE is a local matter.

13.5.3.4 P1 Protocol Elements

13.5.3.4.1 P1 Envelope Protocol Elements

Table 13-11 contains Protocol Elements and their classes.

Tbl. 13-11 P1 protocol elements

Element	Class	Restrictions and Comments
MPDU		
UserMPDU	G	
DeliveryReportMPDU	G	
ProbeMPDU	H	
UserMDPU		
UMPDUEnvelope	M	
UMPDUContent	M	
UMPDUEnvelope		
MPDUIdentifier	M	
originator ORname	M	
originalEncodedInformationTypes	G	If this field is absent, then the Encoded Information Type is "unspecified".
ContentType	M	
UAContentID	H	Maximum length = 16 characters.
Priority	G	
PerMessageFlag	G	Maximum length = 2 octets.
deferredDelivery	X	
PerDomainBilateralInfo	X	No limit on number of occurrences.
RecipientInfo	M	Maximum number = 32K - 1 occurrences. More severe limitations are by bilateral agreement.
TraceInformation	M	
UMPDUContent	M	
MPDUIdentifier		
GlobalDomainIdentifier	M	
IA5String	M	Maximum length = 32 characters, graphical subset only. Refer to T.50 for clarification of graphical subset.
PerMessageFlag		
discloseRecipients	H	
conversionProhibited	G	
alternateRecipientAllowed	H	
contentReturnRequest	X	

(Continued on next page.)

Tbl. 13-11 P1 protocol elements, Continued

Element	Class	Restrictions and Comments
PerDomainBilateralInfo		
CountryName	M	Maximum length = 3 characters.
AdministrationDomainName	M	Maximum length = 16 characters.
BilateralInfo	M	Maximum depth = 8; maximum length = 1024 octets (including encoding).
RecipientInfo		
recipient	M	
ExtensionIdentifier	M	Maximum value = 32K - 1 (32767).
perRecipientFlag	M	Maximum length = 2 octets.
ExplicitConversion	X	
perRecipientFlag		
ResponsibilityFlag	M	
ReportRequest	M	
UserReportRequest	M	
TraceInformation		Reference should be made to Version 5 of the X.400 Implementor's Guide for information related to Trace sequencing.
GlobalDomainIdentifier	M	
DomainSuppliedInfo	M	
DomainSuppliedInfo		
arrival	M	
deferred	X	
action	M	
0=relayed (value)	G	
1=rerouted (value)	H	Rerouting is not required.
converted	H	
previous	H	
ORName		See section 13.5.3.5
EncodedInformationTypes		
bit string	M	Delivery can only occur if match is made with Registered Encoded Information Types. Individual vendors may impose limits. Maximum length = 3 octets.
G3NonBasicParameters	X	
TeletexNonBasicParameters	X	
PresentationCapabilities	X	

Tbl. 13-11 P1 protocol elements, Continued

Element	Class	Restrictions and Comments
DeliveryReportMPDU		
DeliveryReportEnvelope	M	
DeliveryReportContent	M	
DeliveryReportEnvelope		
report	M	
originator	M	
TraceInformation	M	
DeliveryReportContent		
original	M	
intermediate	G	See comment at end of table.
UAContentID	G	
ReportedRecipientInfo	M	Maximum number = 32K - 1 occurrences.
returned	H	Can only be issued if specifically requested in the originating message.
billingInformation	X	Maximum depth = 8; maximum length = 1024 octets (including encoding).
ReportedRecipientInfo		
recipient	M	
ExtensionsIdentifier	M	
PerRecipientFlag	M	
LastTraceInformation	M	
intendedRecipient	H	
SupplementaryInformation	X	Maximum length = 64 characters. Note: This length is subject to change. Value is pending verification by the CCITT SG VIII or IX.
LastTraceInformation		
arrival	M	
converted	G	
Report	M	

Tbl. 13-11 P1 protocol elements (continued from previous page.)

Element	Class	Restrictions and Comments
Report		
DeliveredInfo	G	Generated if delivery is reported.
NondeliveredInfo	G	Generated if failure to deliver is reported.
DeliveredInfo		
delivery	M	
typeofUA	R	This element must be generated with a PRIVATE value by PRMDs.
NonDeliveredInfo		
ReasonCode	M	
DiagnosticCode	H	Whenever possible, use a meaningful diagnostic code.
ProbeEnvelope		
probe	M	
originator	M	
ContentType	M	
UAContentID	H	Maximum length = 16 characters.
original	G	If this field is absent, then the Encoded Information Type is "unspecified".
TraceInformation	M	
PerMessageFlag	G	
contentLength	H	
PerDomainBilateralInfo	X	
RecipientInfo	M	Maximum number = 32K - 1 occurrences.
GlobalDomainIdentifier		
CountryName	M	Maximum length = 3 characters.
AdministrationDomainName (4)	M	Maximum length = 16 characters or digits.
PrivateDomainIdentifier	R	Maximum length = 16 characters or digits. This element must be generated by PRMDs.
End of Definitions		

Notes on Table 13-11

Comment on intermediate TraceInformation in the DeliveryReportContent in table 13-11: Audit and confirmed reports should not be requested by other than the originating domain for two reasons. First, the return path of the report may be different from the path taken by the original message, and it may exclude the domain that added the request for audit and confirmed to the message. Second, if the return path is different from the path of the original message, the originating domain would receive intermediate trace information that it did not request.

13.5.3.5 ORName Protocol Elements

Only form 1 variant 1 O/R names are supported.

Table 13-12 contains ORName protocol elements.

Tbl. 13-12 ORName protocol elements

Element	Class	Restrictions and Comments
ORName		
StandardAttributeList	M	
DomainDefinedAttributeList	X	
StandardAttributeList (1)		
CountryName	R	As defined in X.411, Maximum length = 3 characters.
AdministrationDomainName (4)	R	Maximum length = 16 characters or digits.
X121Address	X	Maximum length = 15 digits.
TerminalID	X	Maximum length = 24 characters.
PrivateDomainName (2)	G	Maximum length = 16 characters.
OrganizationName (2)	G	Maximum length = 64 characters.
UniqueUAIdentifier	X	Maximum length = 32 digits.
PersonalName	G	Maximum length of values of sub-elements = 64 characters. Note: The possibility that this value may be reduced to 40 characters is for further study by the CCITT.
OrganizationalUnit (3)	G	Maximum length = 32 characters per occurrence. A maximum of four occurrences are allowed.
DomainDefinedAttributeList (5)		Maximum = 4 occurrences.
type	M	Maximum length = 8 characters.
value	M	Maximum length = 128 characters.
PersonalName		
surName	M	Maximum length = 40 characters.
givenName	G	Maximum length = 16 characters.
initials	G	Maximum length = 5 characters; excluding surname initial and punctuation and spaces.
generationQualifier	G	Maximum length = 3 characters.

(Continued on next page.)

Tbl. 13-12 ORName Protocol Elements, Continued

Notes:

1. The following apply for comparison of the Standard Attributes of an O/R Name:

   a. Lower case is interpreted as upper case (for IA5).

   b. Multiple spaces may be interpreted as a single space. Originating domains shall only transmit single significant spaces. If multiple spaces are transmitted, non-delivery may occur.

2. At least one of these must be supplied.

3. These should be sent in ascending sequence, from the least significant <Organizational Unit> (lowest in organization hierarchy) to the most significant. Only those specified should be sent. (That is, an unspecified <Organizational Unit> should not be sent along as a field of [null] content, nor zero length, etc.)

4. This attribute shall contain one space in all ORNames of messages originated in a PRMD that is not connected to an ADMD, and in ORNames of recipients reachable only through a PRMD; otherwise, this attribute shall contain an appropriate ADMD name.

5. Many existing mail systems require attributes not present in these agreements. Domain Defined Attributes are a method of providing these. Failure to support the specification of DDAs may prevent successful interworking with such existing mail systems until such time as all mail systems are capable of supporting delivery via the standard attribute list only. Specific recommendations on the use of DDAs are in the Recommended Practices section.

13.5.3.6 P2 Protocol Profile (Based on [X.420])

Tables 13-13 and 13-15 classify the support for the P2 protocol elements required by this profile. The tables give restrictions and comments in addition to X.420.

Restriction on length is one of the types of restrictions. The reaction of implementations to a violation of this restriction is not defined by this Profile.

13.5.3.6.1 P2 Protocol - Heading

Table 13-13 below specifies the support for protocol elements in P2 Headings.

Tbl. 13-13 P2 heading protocol elements

Element	Class	Restrictions and Comments
UAPDU		
IM-UAPDU	G	
SR-UAPDU	X	
IM-UAPDU		
Heading	M	
Body	M	
Heading		
IPMessageId	M	
originator	R	
authorizingUsers	H	
primaryRecipients	G	At least one of primaryRecipients,
copyRecipients	G	copyRecipients, or blindCopyRecipients must be
blindCopyRecipients	H	present.
inReplyTo	G	
obsoletes	H	
crossReferences	H	
subject	G	Maximum length = 256 octets; the ability to generate the maximum size subject is not required.
expiryDate	H	
replyBy	H	
replyToUsers	H	
importance	H	Appropriate action is for further study.
sensitivity	H	Appropriate action is for further study.
autoforwarded	H	

(Continued on next page.)

Tbl. 13-13 P2 heading protocol elements, continued

Element	Class	Restrictions and Comments
IPmessageId		
ORName	H	
PrintableString	M	Maximum length = 64 characters.
ORDescriptor		
ORName	H	Specify the ORName whenever it is possible. See Appendix 13B.
freeformName	H	Maximum length = 64 characters, graphical subset only (128 octets.)
telephoneNumber	H	Maximum length = 32 characters. This allows for punctuation. It does not take into account possible future use by ISDN.
Recipient		
ORDescriptor	M	
reportRequest	X	
replyRequest	H	
Body		No limit on number of BodyParts.
BodyPart	G	No limit on length of any BodyPart or the depth of ForwardedIPMessage BodyParts nested. Classification is subject to pending CCITT resolution
SR-UAPDU		
nonReceipt	H	
receipt	H	
reported	M	
actualRecipient	R	
intendedRecipient	H	
converted	X	
NonReceiptInformation		
reason	M	
nonReceiptQualifier	H	
comments	H	Maximum length = 256 characters.
returned	H	May only be issued if specifically requested by originator.

(Continued on next page.)

Tbl. 13-13 P2 heading protocol elements, continued

Element	Class	Restrictions and Comments
ReceiptInformation		
receipt	M	
typeOfReceipt	H	
SupplementaryInformation	X	Maximum length = 64 characters. Note: This value is pending verification by the CCITT SG VIII or IX.
End of Definitions		

13.5.3.6.2 P2 Protocol - BodyParts

a) All BodyParts with identifiers in the range 0 up to and including 16K -1 are legal and should be relayed. BodyPart identifiers corresponding to X.121 Country Codes should be interpreted as described in Note 2 of figure 13-14.

- o Implementations are required to generate and image IA5Text.
- o Implementations should specify the other BodyPart types supported.
- o If an implementation supports a particular BodyPart type for reception, it should also be able to support that BodyPart type for reception if it is part of a ForwardedIPMessage.
- o For the BodyPart types currently considered, support for the protocol elements is as indicated in table 13-15.

b) Privately Defined BodyParts

This section describes an interim means for identifying privately defined BodyParts. This section shall be replaced in a future version taking into account CCITT recommendations with equivalent functionality.

```
BodyPart  :: =  CHOICE {
   [0]IMPLICIT IA5Text,
   [1]IMPLICIT TLX,
         .
         .
         .
   [234]IMPLICIT UKBodyParts,
         .
         .
         .
   [310]IMPLICIT USABodyParts,
         .
         .
         .                          }
   Where UKBodyParts and USABodyParts are defined as:

     SEQUENCE {BodyPartNumber, ANY}

     BodyPartNumber ::= INTEGER
```

Note 1: In the EncodedInformationTypes of the P1 Envelope, the undefined bit must be set when a message contains a privately defined BodyPart. Each UA that expects such BodyParts should include undefined in the set of deliverable EncodedInformationTypes it registers with the MTA.

Note 2: All BodyPartNumbers assigned must be interpreted relative to the BodyPart in which they are used, which is that tagged with the value [310] for those defined within the United States. The NBS assigns unique message BodyPartNumbers for privately defined formats within the United States.

Fig. 13-14 X.409 Definition of Privately Defined BodyParts

13.5.3.6.3 P2 BodyPart Protocol Elements

Tbl. 13-15 P2 BodyParts

Elements	Class	Restrictions and Comments
BodyPart		
IA5Text	G	
TLX	X	
Voice	X	
G3Fax	X	
TIFO	X	
TTX	X	
Videotex	X	
NationallyDefined	X	
Encrypted	X	
ForwardedIPMessage	H	
SFD	X	
TIF1	X	
unidentified	X	
IA5Text		
repertoire	H	
IA5String	M	For rendition of IA5Text see Appendix 13C.
TLX		For further study by CCITT.
Voice		
Set		For further study by CCITT.
BitString	M	
G3Fax		
numberOfPages	X	
P1.G3NonBasicParameters	X	
SEQUENCE (OF BIT STRING)	M	
BIT STRING	H	See Note.
P1.G3NonBasicParameters		Support for individual elements is for further study.
TIFO		
T.73Document	M	
T.73ProtocolElement	H	See Note.

(Continued on next page.)

Tbl. 13-15 P2 BodyParts, continued

Elements	Class	Restrictions and Comments
TTX		
numberOfPages	X	
telexCompatible	X	
P1.TeletexNonBasicParams	X	
SEQUENCE	M	
T61String	H	See Note.
P1.TeletexNonBasicParams		
graphicCharacterSets	X	
controlCharacterSets	X	
pageFormats	X	
miscTerminalCapabilities	X	
privateUse	X	
Videotex		
SET		For further study by CCITT.
VideotexString	M	
NationallyDefined		
ANY	M	
Encrypted		
SET		For further study by CCITT.
BIT STRING	M	
ForwardedIPMessage		
delivery	H	
DeliveryInformation	H	
IM-UAPDU	M	
DeliveryInformation		
P1.ContentType	M	
originator	M	
original	M	
P1.Priority	G	
DeliveryFlags	M	
otherRecipients	H	
thisRecipient	M	
intendedRecipient	H	
converted	X	
submission	M	

(Continued on next page.)

Tbl. 13-15 P2 BodyParts, continued

Elements	Class	Restrictions and Comments
SFD		
SFD.Document	M	
TIF1		
T73 Document	M	
T73.ProtocolElement	H	See note.

Note: This element is not an addition to the definition of the BodyPart. It is described here to show that the SEQUENCE may contain zero elements. A Problem Report has been submitted to the CCITT to clarify whether this is permissible. The NBS/OSI Workshop will adopt the CCITT decision.

13.5.4 Reliable Transfer Server (RTS)

13.5.4.1 Implementation Strategy

Based on X.410 clause 3 and X.411 clause 3.5.

13.5.4.2 RTS option selection

a) The maximum number of simultaneous associations is not limited in this profile; if the capacity of a system is exceeded, it should not initiate or accept additional associations.

b) Associations are established by the MTA which has messages to transfer.

c) Associations are released when they are not needed. Associations may also be ended prematurely due to internal problems of the RTS.

d) For both monologue and two way alternate associations, the initiator keeps the initial turn.

e) When establishing an RTS association, the following rules apply to the use of parameters in addition to those in X.410 clause 3.2.1:

Dialogue mode: Monologue must be supported for this profile; two-way alternate is used only if both partners agree.

Initial turn: Kept by the initiator of the association.

f) The 'priority-mechanism' and the 'transfer-time limit' are regarded as local matters.

13.5.4.3 RTS Protocol Options and Clarifications

Realization of the RTS protocol is subject to the following rules in addition to those specified in X.410 clause 4:

a) One RTS association corresponds to one or more consecutive session connections (not concurrent ones). The first is opened with ConnectionData of type OPEN, and subsequent ones are opened with type RECOVER.

b) Recovery of a Session connection is only by RTS initiator.

c) Checkpoint size:

- o Checkpointing and No Checkpointing should be supported. Whenever possible, checkpointing should be used.
- o The minimum checkpointSize is 1 (that is, 1024 octets).

d) Window size:

- o Minimal value of 1 (if checkpointing is supported).
- o WindowSize = 1 means: After an S-SYNCH-MINOR request is sent, wait until the confirmation is received before issuing an S-DATA, S-SYNCH-MINOR, or S-ACTIVITY-END request.

e) APDUs should not be blocked into one activity.

f) Only one SSDU shall be transferred:

- o Between two adjacent minor synch points.
- o Between minor synch points and adjacent S-ACTIVITY-START and S-ACTIVITY-END requests.
- o Between S-ACTIVITY-START and S-ACTIVITY-END without checkpoints.

g) A monologue association is defined as follows:

- o The RTS user responsible for establishing the association is called the initiator.
- o The initiator keeps the initial turn.

- o APDUs are transferred in the direction of the initiator to the recipient only.

- o There shall be no token passing.

- o Only the initiator can effect an orderly release of the association.

h) A two-way alternate session is as described in X.410.

i) In the UserData parameter of the S-U-ABORT, the ReflectedParameter will not be used in the AbortInformation element.

j) When the S-ACTIVITY-RESUME is used to resume an activity in the same session connection as the one in which it started, this must happen immediately after the activity has been interrupted (i.e., no intervening activity can occur). Otherwise, X.410 clause 4.3 paragraph 1 may be violated.

k) When S-ACTIVITY-RESUME is used to resume an activity started in another session connection, the following conditions must be met:

- o The current session connection is of type "recover".

- o The value of OldSessionConnectionIdentifier in S-ACTIVITY-RESUME must match the value of the SessionConnectionIdentifier parameter used in the S-CONNECT of the prior session connection. This value is also identical to the SessionConnectionIdentifier in the ConnectionData (in PConnect, in SS-UserData) for the current session connection.

- o This must occur as the first activity of the next session connection for the same RTS-association. It must be the first, otherwise X.410 clause 4.5.1 point 1 is violated.

Note: It is in the same RTS-ASSOCIATION because the use of S-ACTIVITY-RESUME only makes sense within the scope of one RTS association.

l) If the transfer of an APDU is interrupted before the confirmation of the first checkpoint, the value of the SynchronizationPointSerialNumber in S-ACTIVITY-RESUME should be zero, and the S-ACTIVITY-RESUME must be immediately followed by an S-ACTIVITY-DISCARD.

m) In S-TOKEN-PLEASE, the UserData parameter shall contain an integer conforming to X.409 which conveys the priority.

n) The receiving RTS can use the value of the Reason parameter in

the S-U-EXCEPTION-REPORT to suggest to the sending RTS that it should either interrupt or discard the current activity.
As stated in version 5 of the X.400 Series Implementor's Guide, "On receipt of an 'unrecoverable procedure error' the current activity is not recoverable and the sending RTS issues an S-ACTIVITY-DISCARD. On receipt of any other reason code (including a nonspecific error), the sending RTS issues an S-ACTIVITY-INTERRUPT followed by an S-ACTIVITY-RESUME."

o) In the case of S-P-ABORT, the current activity (if any) is regarded as interrupted, rather than discarded.

p) Table 13-16 illustrates the legal negotiation possibilities allowed by X.410 clause 4.2.1 regarding checkpoint size and window size.

Tbl. 13-16 Checkpoint window size of IP

		acceptor answer		
		CS = 0 (or unspecified) WS unspecified	CS = m WS = j (or unspecified)	CS = n WS = j (or unspecified)
initiator proposal	CS = 0 (or unspecified) WS = i (or unspecified)	legal	legal	legal
	CS = k WS = i (or unspecified)	legal	legal	not allowed

Legend:

- o CS means CheckpointSize
- o WS means WindowSize
- o i, j, k, m, and n are integer values with the following relations:

  $0 < m < k < n$ (values assigned to CS)
  $0 < j < i$ (values assigned to WS)

- o For unspecified parameters, the default applies. In this case, the numeric relations apply, that is, the default values substitute for the unspecified integer.

13.5.4.4 RTS Protocol Limitations

The RTS Protocol Limitations for this profile are listed in table 13-17.

Tbl. 13-17 RTS protocol elements

Element	Class	Restriction
PConnect	M	
DataTransferSyntax	M	Value = 0.
pUserData	M	
checkpointSize	H	
windowSize	H	
dialogueMode	H	
ConnectionData	M	
applicationProtocol	R	Value = 1.
ConnectionData		
open	G	
recover	G	
open		
RTS user data	G	
recover		
SessionConnectionIdentifier	G	
RTS user data		
mTAName	G	Maximum length 32 characters graphic subset of IA5 only.
password	G	Maximum length 64 octets graphic subset of IA5 only.
< null RTS User Data >	G	Generated if other validation methods are used.
SessionConnectionIdentifier		
CallingSSUserReference	M	Maximum length 64 octets including encoding = 62 octets of T.61.
CommonReference	M	
AdditionalReferenceInformation	H	Maximum length 4 octets including encoding = 2 octets of T.61.
PAccept	G	
DataTransferSyntax	M	Value = 0.
pUserData	M	
checkpointSize	H	
windowSize	H	
ConnectionData	M	

(Continued on next page.)

Tbl. 13-17 RTS protocol elements, continued

Element	Class	Restriction
PRefuse	G	
RefuseReason	M	
SS User Data (in S-TOKEN-PLEASE)	G	
AbortInformation (in S-U-ABORT)	G	
AbortReason	H	
reflectedParameter	X	Restricted to 8 bits.
End of Definitions		

13.5.5 Use of Session Services

The session requirements and use of session are covered in section 8 of this document.

13.5.6 Data Transfer Syntax

This section defines Presentation Transfer Syntax and notation rules applicable to these agreements. Implementations must conform EXACTLY as specified in X.409 with no further restrictions. Appendix 13C defines rendition of IA5 Text and T61 characters.

13.6 PRMD to ADMD and ADMD to ADMD

13.6.1 Introduction

This section defines the implementation agreements that apply to the interface between two management domains when at least one is an ADMD. A message arriving at an ADMD has either no recipient within that domain or one or more recipients within that domain. In the former case, the ADMD serves as a relay between two or more domains and the actions required of that ADMD are independent of the nature (PRMD or ADMD) of the domains. In the latter case, the ADMD is responsible for delivering messages to the proper recipient(s) within its jurisdiction, and may also be responsible for relaying the message.

Given the two roles for an ADMD, this section describes two distinct sets of functional requirements for an ADMD. The first is the

relaying requirement that is needed to provide PRMD and other ADMD interworking. The second is analogous to the PRMD's support to its customers through the integrated UAs. These are distinct functional differences. The services provided to the UAs of an ADMD are independent of the requirement that an ADMD provide a function for interworking with any type of Management Domain (MD). Figure 13-18 illustrates the two roles played by an ADMD.

This section is presented in the form of deviations from the agreements applicable to PRMD-to-PRMD (section 13.5). Unless explicitly noted in the remainder of this section, all of the specifications for PRMD to PRMD apply to PRMD to ADMD and ADMD to ADMD.

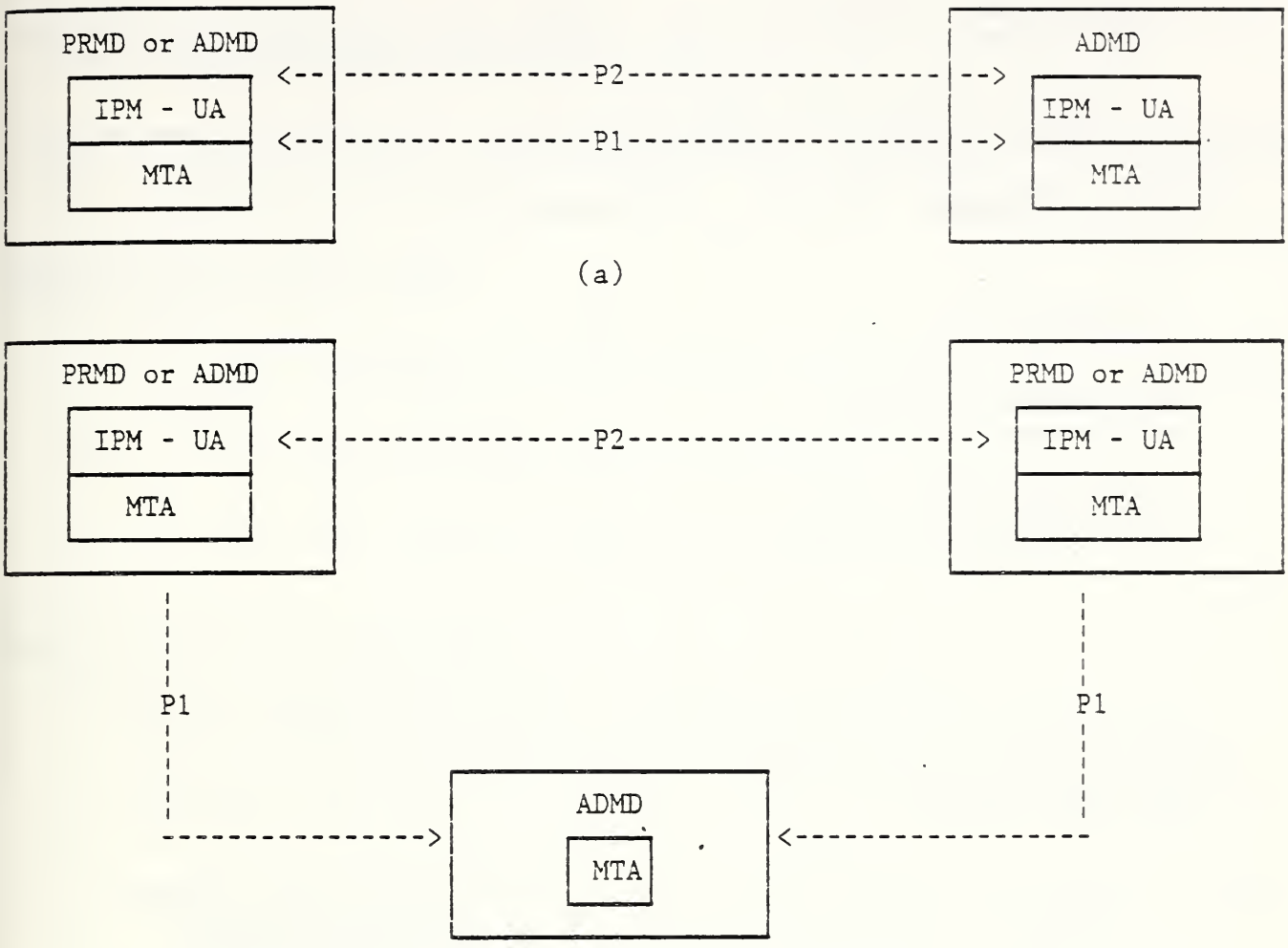

Fig. 13-18 An ADMD may (b) or may not (a) serve as a relay.

13.6.2 Additional ADMD Functionality

The following defines the additional ADMD specific functionality required over and above that specified in the PRMD section.

13.6.2.1 Relay Responsibilities of an ADMD

ADMDs will relay all content types (not just P2) unchanged in the absence of a request for conversion.

13.6.2.2 P1 Protocol Classification Changes

Table 13-19 describes the changes to the PRMD P1 Protocol classifications required for a delivering Administration domain (with respect to the original message; this means the domain which originates the delivery reports).

Protocol Elements	Class	
DeliveredInfo		
typeOfUA	H	
ReportedRecipientInfo		
SupplementaryInformation	H	See Note 1.
GlobalDomainIdentifier		
PrivateDomainIdentifier	H	

For relaying Administration domains, the classifications are all "X"

For originating Administration domains, these are all "NOT APPLICABLE".

Note 1: Domains providing access to TELEX/TELETEX recipients, whether directly or indirectly as a result of bilateral agreements between domains, must ensure that this information, when present, is accessible by the recipient of the delivery report.

Tbl. 13-19 P1 Protocol Classification Changes for a Delivering ADMD

13.6.2.3 O/R Names

O/R Names shall consist of:

- CountryName,
- AdministrationDomainName.

as well as one of the following:

- PrivateDomainName,
- PersonalName,
- OrganizationName,
- OrganizationalUnit,
- UniqueUAIdentifier,

- o X121Address.

and permits the optional inclusion of a
- o DomainDefinedAttributeList.

Note: The destination PrivateDomainName or OrganizationName must be present if destined for a PRMD. The ADMD relaying the message to that destination PRMD requires this element.

13.6.2.4 P1 Originator Name

Management Domains (MDs) must specify in the ADMD name field of the O/R Name StandardAttributeList in P1, the name of the Administration domain:

- o to which the message is being sent (in recipient names)
- o from which the message originated (in the originator name).

13.6.3 Interworking with Integrated UAs

If the message originates at a UA owned by an ADMD, or is delivered to such a UA, the O/R Name follows the same Form 1 Variant 1 constraints as the base specifications; except that the ADMD name is the name of the ADMD that owns the UA and instead of supplying a PRMD Name, one (or more) of the following must be provided:

- o OrganizationName,
- o OrganizationalUnit,
- o PersonalName.

and may optionally include a

- o DomainDefinedAttributeList.

13.6.4 Differences with Other Profiles

13.6.4.1 NTT Profile

There are no outstanding issues regarding interworking between NTT-conformant systems and NBS-conformant systems with the exception of the number of recipients. The ExtensionIdentifier field may contain a maximum value of 32K-1; however, according to the current NTT profile, if a message with more than 256 recipients is received, the NTT-conformant domain will generate a nondelivery notification. This also applies to the ReportedRecipientInfo in a delivery report.

13.6.4.2 CEPT Profile

See Appendix 13E.

13.6.5 Connection of PRMDs to Multiple ADMDs

Given that Management Domain names (both PRMD and ADMD) shall be unique within the U.S., then when an ADMD is presented a message for transfer from a PRMD, it will accept O/R Names (both originator and recipient) which have an AdministrationDomainName field value different than the Administration's name. "Accept" implies the attempt to route/deliver the message shall be made, as appropriate, based upon the knowledge that MD names are unique.

Whether this functionality is required by an Administration for conformance to this agreement is for further study.

If a PRMD is connected to two or more ADMDs which are not effectively connected (either directly or via a third ADMD), full X.400 functionality shall not be available. Problems occur especially in the areas of:

- o Naming,
- o Routing,
- o Replying.

13.6.6 Connection of an ADMD to a Routing PRMD

It is possible for a collection of interconnected private domains to establish one domain as the "gateway" to an ADMD, and hence to the world.

If an ADMD is connected to such a gateway PRMD, the individual private domains shall be registered with the Administration. Administrations need not support such connections.

Note also that upon receipt by the ADMD of a message originating somewhere within the PRMD collection, that the TraceInformation may contain more than one element.

The X.400 Recommendations specify that an ADMD should not attempt to relay a message destined for another ADMD through a PRMD, thus an ADMD should ensure that messages destined for another ADMD are not relayed through a PRMD. It should be noted, however, that a relaying PRMD will relay any such message it receives.

13.6.7 Management Domain Names

- All Management Domain Names (both Private and Administration) shall be unique within the U.S.

- A central naming authority shall be established to register domain names.

13.6.8 Envelope Validation Errors

For validation errors, a non-delivery notice shall be generated (if possible) with reason code of 'unableToTransfer' and diagnostic code of 'invalidParameters' (unless specified otherwise).

ADMDs will validate P1 Envelopes in the following areas:

a) The X.409 syntax of all elements should be checked.

b) The pragmatic constraint limits (lengths of fields and number of occurrences of fields) should be checked.

c) Semantic validation of the following elements should be done:

   - originator O/R Name,
   - recipient O/R Name in the RecipientInfo,
   - Priority.

   Only recipient Names with the responsibility flag set should be validated. The validation of O/R names is defined in 13.8.3.3; the validation of priority is defined in 13.8.3.7.1.

d) MPDU Identifier Validation

   - Validation of the GlobalDomainIdentifier component of the MPDU Identifier is performed upon reception of a message (i.e., as a result of a TRANSFER.Indication).

   - The country name should be known to the validating domain, and depending on the country name, validation of the ADMD name may also be possible.

   - Additional validation of the GlobalDomainIdentifier is performed against the corresponding first entry in the TraceInformation. If inconsistencies are found during the comparison, a non-delivery notice with the above defined reason and diagnostic codes is generated.

   - A request will be generated to the CCITT for a more meaningful diagnostic code (such as 'InconsistentMPDUIdentifier').

13.6.9 Quality of Service

13.6.9.1 Domain Availability

13.6.9.1.1 ADMD Availability

The goal is to provide 24 hour per day availability. Note that there will be periods of time when an ADMD may be unavailable due to maintenance windows in its supporting network or in an MTA within the domain.

13.6.9.1.2 PRMD Availability

Although the goal of PRMD availability is also 24 hours per day, business reasons are likely to dictate some different level of availability. ADMDs shall require a profile from the PRMD that indicates its schedule of regular availability to the ADMD.

13.6.9.2 Delivery Times

In the absence of standardized quality of service parameters, the following are agreed to. When standardized parameters from CCITT Study Group I become available, they shall be adopted.

a) In table 13-20 the following delivery time targets are established:

Delivery Class	95% Delivered Before
Urgent	3/4 hour
Normal	4 hours
Non-Urgent	24 hours

Tbl. 13-20 Delivery Time Targets

b) The interval(s) between retries and the number of retry attempts that an ADMD uses in attempting delivery to a PRMD or integrated UA, will be locally determined domain parameters. However, the total elapsed times after which delivery attempts will be stopped are shown in table 13-21. This implies that, after these times, a Non-Delivery Notice will be generated.

An Administration shall continue to attempt delivery until the forced nondelivery time, even if the recipient domain has scheduled an unavailability window.

Delivery Class	NonDelivery Forced After
Urgent	4 hours
Normal	24 hours
Non-Urgent	36 hours

Tbl. 13-21 Forced Nondelivery Times

Note: Both tables apply to the period between acceptance by the originating MTA in the originating Administration domain to the time of delivery in the destination Administration domain. Transit time within PRMDs is NOT included in the above times.

13.6.10 Billing Information

a) All aspects relating to billing, charging, tariffs, settlement, and in particular to the use of the billingInformation field in the delivery report, is subject to bilateral agreement, and shall not be addressed in these implementation agreements.

b) No ADMD shall require a PRMD to supply or process billing information.

13.6.11 Transparency

a) No P1 extensions are to be allowed. Should an ADMD receive a message containing P1 extensions, it shall generate a non-delivery notice (if possible) with reason code of unableToTransfer and diagnostic code of invalidParameters.

b) The CCITT has been requested to establish a more meaningful diagnostic code (such as protocolError) for this occurrence.

c) P2 extensions shall be relayed transparently by ADMDs.

13.6.12 RTS Password Management

RTS password management shall be a local matter. This includes:

- password length
- frequency of changes
- exchange of passwords with communicating partners
- loading passwords (i.e., the timing of password changes with respect to active associations).

13.6.13 For Further Study

Issues requiring further study are:

- o Intra-Domain Routing
- o Multi-Vendor Domains

13.7 INTER and INTRA PRMD CONNECTIONS

13.7.1 Introduction

This section is limited in scope to issues arising from the indirect connection of a PRMD to another PRMD or to an ADMD, and to the interconnection of MTAs to form inter-PRMD connections. Indirect means that the connection is made via a relaying PRMD. The X.400 Recommendations describe the way that a PRMD connects to a ADMD and the way that an ADMD connects to another ADMD. The Recommendations do not, however, describe the way that a PRMD connects indirectly to an ADMD or another PRMD, nor do they describe the way that MTAs are connected within a PRMD. These configurations (shown in Figures 13-22 and 13-23) are useful, for example, in connecting equipment from different vendors at a single customer site.

The P1 protocol and its related services for both inter and intra PRMD connections are addressed in this section. In addition, a method for routing within a PRMD is given. It is recognized that uniform methods for Administration, maintenance, and quality of service should be developed for such configurations, and this work is for further study.

This section describes the minimum that must be provided in order to implement a relaying PRMD and a MTA within a PRMD.

This section is presented in the form of deviations from agreements applicable to PRMD to PRMD connection (section 13.5). That is, unless specifically noted in the remainder of this section, the agreements in section 13.5 apply to both relaying PRMDs and MTAs within a PRMD.

13.7.2 The Relaying PRMD

A PRMD that has the capability of relaying messages to another PRMD is called a relaying PRMD. A PRMD implementation need not claim to be a relaying PRMD. A PRMD implementation which does claim to be a relaying PRMD must follow the implementation agreements in this section.

13.7.2.1 Relay Responsibilities of a PRMD

The responsibilities of a relaying PRMD are the same as those of an ADMD (as specified in sections 13.6.8 and 13.6.2.1). In addition, the PRMD will simply deliver messages routed to it from an ADMD, even if this results in routing a message from the ADMD to the PRMD to another ADMD.

13.7.2.2 Interaction with an ADMD

In order for an ADMD to route a message to ADMD A via ADMD B, it must know that A is reachable through B. Similarly, in order for any MD to route a message to PRMD A via a relaying PRMD B, it must know that A is reachable through B (see Figure 13-24).

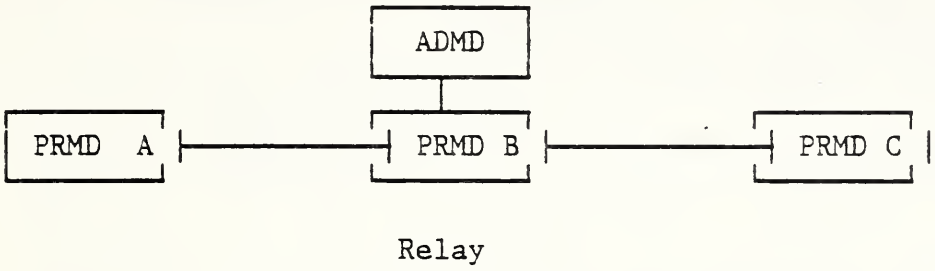

Fig. 13-22 Relaying PRMD

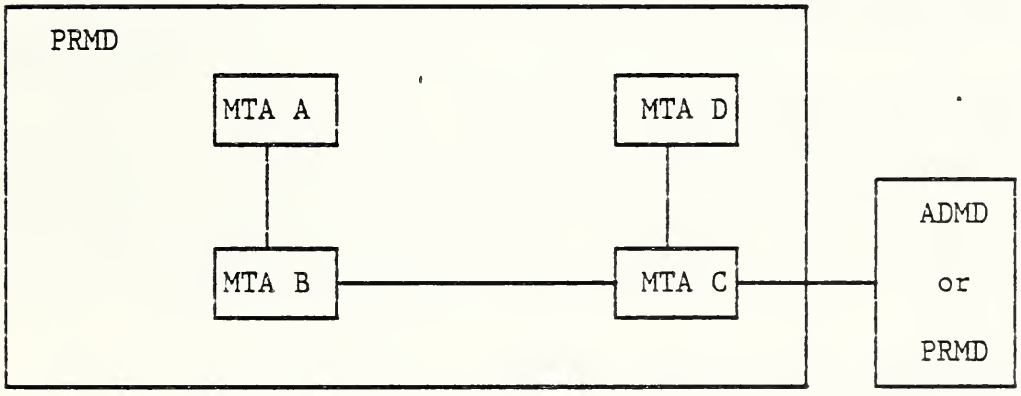

Fig. 13-23 Intra PRMD connections

Note 1: Section 13.6.6 specifies that ADMDs are not required to connect to a relaying PRMD, but they are not precluded from doing so.

Note 2: TraceInformation may have more than one sequence on submission of a message by a relaying PRMD to an ADMD.

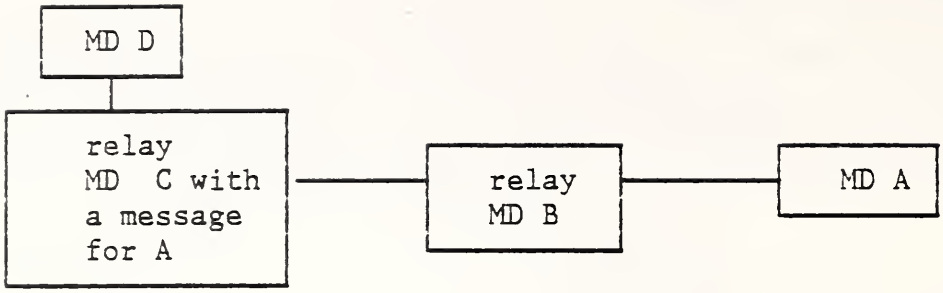

Fig. 13-24 MD C must know of A to route the message

13.7.3 Intra PRMD Connections

A PRMD is composed of MTAs which cooperate to perform the functions expected of a domain. An MTA implementation need not claim to follow the implementation agreements of this section.

13.7.3.1 Relay Responsibilities of an MTA

The relaying responsibilities of an MTA are the same as those of an ADMD (as specified in sections 13.6.8 and 13.6.2.1) with one exception: loop suppression within the domain is done using the MOTIS InternalTraceInfo protocol element. The MTA must validate the InternalTraceInfo (see section 13.8.3.5 for details on validation). In addition, the PRMD will simply deliver messages routed to it from an ADMD, even if this results in routing a message from the ADMD to the PRMD to another ADMD (please see section 13.6.6).

13.7.3.2 Loop Suppression within a PRMD

a) The only mechanism define in the X.400 Recommendations for suppressing loops is TraceInformation, which is added on a per domain basis to detect and suppress loops among domains. Loops among MTAs within a domain need to be detected and suppressed. This implies that each MTA must add trace information that is meaningful within the domain. The MOTIS solution of adding InternalTraceInfo to the P1 Envelope of a message was adopted. The definition of InternalTraceInfo is given in table 13-25. The InternalTraceInfo is added by each MTA within a PRMD to handle a message, and it is examined in the same way as TraceInformation to detect and suppress loops.

```
InternalTraceInfo ::= [APPLICATION 30]
  IMPLICIT SEQUENCE OF
  SEQUENCE {
     MTAName,
     MTASuppliedInfo }

MTAName ::= PrintableString
```

Fig. 13-25 Definition of InternalTraceInfo

If the MTAName and password of X.411 are used for validation, then it is recommended that the MTAName used for validation also be used in the InternalTraceInfo. However, there is a complication: in X.411, MTAName is an IA5String, and the MTAname defined by MOTIS is a PrintableString. Efforts will be made to change the MOTIS definition from PrintableString to IA5String.

b) Three actions are defined in MTASuppliedInfo: relayed, rerouted, and recipientReassignment as shown in table 13-26. The recipientReassignment action is not supported in these agreements. The ability to generate it is not required, and if it is present on an incoming message, the action field will be ignored.

```
MTASuppliedInfo  ::= SET {
  arrival [0] IMPLICIT Time,
  deferred [1] IMPLICIT Time OPTIONAL,
  action [2] IMPLICIT INTEGER
     { relayed(0), rerouted(1), recipientReassignment(2) }
  previous MTAName OPTIONAL }
```

Fig. 13-26 Defined Actions in MTASuppliedInfo

13.7.3.3 Routing Within a PRMD

a) Routing within a PRMD is complicated by the lack of a directory standard. In particular, it constrains intra-domain routing decisions to be based on some combination of the intra-domain attributes of the O/R Name, Organization Name Organizational Units, and Personal Name. In order to enhance interworking and to reduce the difficulty of configuring intra-domain connections, it is useful to restrict the ways in which these may be used in making routing decisions.

b) However, it is recognized that vendors may wish to provide MTAs with varying degrees of routing capability within a PRMD as a temporary expedient until appropriate standards for automated construction of directories and routing tables are available. This section assigns class numbers to certain levels of routing capability and discusses the consequences of using MTAs which fall into each class. The classification scheme will allow some diversity in allocating O/R Name space and in configuring intra-domain routes.

c) When other methods are recommended by standards bodies, the classification scheme described here will become obsolete. Large-scale, multi-vendor PRMDs may not be practical in the absence of standardized methods.

13.7.3.3.1 Class Designations

When it is clear that a message is to be delivered within a domain, the Country Name, ADMD Name, and PRMD Name have already served their purpose in determining the next MTA in the route to the recipient. The remaining fields that might be used on making routing decisions within the PRMD are the Organization Name, Organizational Units, and Personal Name.

MTAs are classified by their ability to discriminate between O/R names when making routing decisions within a PRMD. Conformant MTAs will be classified as shown in table 13-27.

	Class 1	Class 2	Class 3
Organization Name	H	H	H
SEQUENCE OF Organizational Unit	X	H	H
Personal Name	X	X	H

Tbl. 13-27 Conformant MTA Classifications

a) An 'H' means that the MTA must be able to base its intra-domain routing decisions on the given component of the O/R Name. In particular, both Class 2 and Class 3 MTAs must be able to discriminate on all the members in a supplied sequence of OrganizationalUnits. A Class 3 MTA must be able to discriminate on all of the elements in a PersonalName.

An 'X' means that the MTA need not have the ability to discriminate on the given component.

b) There is a hierarchy in support of components. The ability to discriminate on a given component does not imply the requirement to do so: e.g., a Class 3 MTA is not required to have tables for every PersonalName in the domain. Equally, an MTA which can discriminate on OrganizationalUnits to make routing decisions need not always use the full sequence in an O/R Name if a partial sequence provides enough information.

c) The above classifications only apply to routing decisions in selecting a next hop within a domain. All MTAs are entitled to examine the full O/R Name when identifying their own directly served UAs.

d) The routing table of a Class 1 MTA will be relatively small, because intra-domain routing decisions are based solely on OrganizationName. The routing table of a Class 2 MTA may be substantially larger and more complex because of its ability to discriminate on OrganizationalUnits as well as OrganizationName to make routing decisions. The routing table of a Class 3 MTA may be larger still, because its use of the components of PersonalName in addition to the other information.

13.7.3.3.2 Specification of MTA Classes

If an MTA implementation claims to follow the implementation agreements, it must be either a Class 1, Class 2, or a Class 3 MTA. The class of an MTA implementation should be specified so that PRMD administrators can choose equipment properly.

13.7.3.3.3 Consequences of Using Certain Classes of MTAs

Definition: An MTA which accepts submission requests and furnishes delivery indications to a UA is said to "directly serve" the UA.

a) The presence in a domain of an MTA acting as a Class 1 or Class 2 MTA imposes administrative restrictions on the assignment of O/R Names to UAs and in the configuration of routes within that domain.

- o A Class 1 MTA may directly serve UAs from several OrganizationNames. However, if a Class 1 MTA directly serves a UA with a given OrganizationName, no other MTA in the domain may directly serve a user with the same OrganizationName. This means that if all MTAs in a domain are Class 1, then all UAs with a given OrganizationName must be assigned to the same MTA.

- o A Class 2 MTA may directly serve UAs from any combination of OrganizationName and sequence of OrganizationalUnits. However, if a Class 2 MTA directly serves a UA with a given

combination, no other MTA in the domain may directly serve a user with the same combination. This means that if all MTAs in a domain are Class 2, then all UAs with a given OrganizationName and sequence of OrganizationalUnits must be assigned to the same MTA.

- o A domain consisting entirely of Class 3 MTAs is free of all the above restrictions.

b) If Class 1 or Class 2 MTAs are used to perform relaying within a PRMD containing MTAs of other classes, care must be exercised in determining the topology of the domain to avoid leaving certain UAs inacessible from certain MTAs within the domain. The example below shows one of the configurations that should be avoided. The example is intended to stimulate careful examination of the relationship between network and organizational topologies.

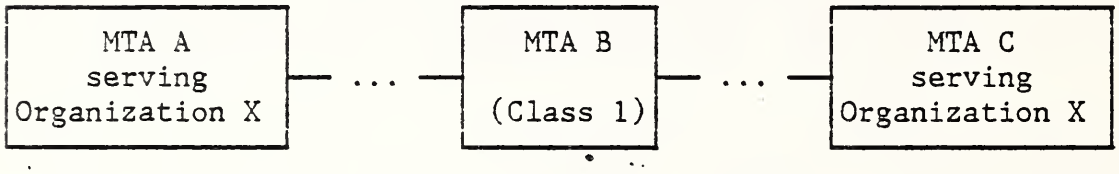

Fig. 13-28 Example of a confirguration to be avoided

In Figure 13-28, B will route all messages for Organization X to either A or C because B is a Class 1 MTA. The administrator who created this configuration probably wanted B to route some messages for Organization X to A, and some to C. However, B does not have the capability for this because it is only a Class 1 MTA. The configuration in Figure 13-28 can be corrected by replacing B with a Class 2 or Class 3 MTA.

13.7.3.4 Uniqueness of MPDUidentifiers Within a PRMD

When generating an IA5String in an MPDUIdentifier, each MTA in a domain must ensure that the string is unique within the domain. This shall be done by providing an MTA designator with a length of 12 octets which is unique within the domain, to be concatenated to a per message string with maximum length of 20 octets.

Two pieces of information, the MTA name and MTA designator, need to be registered within a PRMD to guarantee uniqueness. This registration facility need not be automated. If the MTA name is

less than or equal to 12 characters, it is recommended that it also be used as the MTA designator.

13.7.4 Service Elements and Optional User Facilities

A PRMD made up of MTAs which support varying sets of service elements in addition to those required in these agreements may appear to provide inconsistent service for these elements. For example, if one MTA supports deferred delivery and another MTA does not, then deferred delivery can be used by some, but not all, users in the PRMD. Similarly, if one MTA supports return of contents and another does not, then a user outside of the PRMD will receive returned contents for messages sent to one user, but not for messages sent to another user. Note that this same inconsistency occurs when sending to two PRMDs which support different additional optional elements.

13.7.5 X.400 Protocol Definitions

This section describes additions and modifications to section 13.5.3 which are required for implementation of a relaying PRMD or an MTA within a PRMD.

13.7.5.1 Protocol Classification

a) The classification scheme given in section 13.5.3.1 applies to elements passing from one PRMD to another. For both relaying PRMDs, and MTAs in a PRMD, the same classification scheme will be used, but within a PRMD the classification applies to elements passing from one MTA to another.

b) In addition to the classifications given in section 13.5.3.1, a classification of Prohibited has been used.

PROHIBITED = P

This element shall not be used. Presence of this element is a protocol violation.

13.7.5.2 P1 Protocol Elements

Table 13-29 contains protocol elements and their classes. An * signifies that the classification of the protocol element has not changed from Table 13-11.

Tbl. 13-29 P1 Protocol Elements

Element	Class	Restrictions and Comments
UMPDUEnvelope		
MPDUIdentifier	M*	This field needs to be unique within a PRMD. See sections 13.7.3.4 for the method of ensuring uniqueness.
originator	M*	It is recommended that all components of the originator's ORName be included to help ensure that reports can be returned.
TraceInformation	M*	The first MTA in the domain to receive the message adds the TraceInformation. Subsequent MTAs can update the TraceInformation in the event of conversion or deferred delivery. When a message is generated, the originating MTA adds the TraceInformation.
InternalTraceInfo	M/P	This element is mandatory for envelopes transferred between MTAs within a PRMD, and prohibited in messages transferred outside the domain. Elements are always added to the end of the sequence.
InternalTraceInfo	M	MTANames within a PRMD must be unique. See section 13.7.3.4 for the method of assuring uniqueness Maximum length = 32 characters.
MTAName		
MTASuppliedInfo	M	

(Continued on next page.)

Tbl.13-29 P1 Protocol Elements, continued

Element	Class	Restrictions and Comments
MTASuppliedInfo		
arrival	M	
deferred	X	This field must be generated by MTAs which perform deferred delivery.
action	M	See section 13.7.3.2 for restrictions on values of this field.
previous	X	This field must be generated by MTAs which perform rerouting.
DeliveryReportEnvelope		
TraceInformation	M*	The first MTA in the domain to receive the report adds the TraceInformation. When a report is generated, the originating MTA adds the TraceInformation.
InternalTraceInfo	M/P	This field is mandatory for envelopes transferred between MTAs within a PRMD, and prohibited in messages transferred outside the domain.
DeliveryReportContent		
intermediate InternalTraceInfo	P	If it were possible to include this field in the delivery report content, an audit and confirmed report could be provided to detect problems within a PRMD. Efforts are being made to add this field to the MOTIS definition.
DeliveredInfo		
typeOFUA	R*	It is the responsibility of the MTA generating the report to generate this element.

(Continued on next page.)

Tbl. 13-19 P1 Protocol Elements, continued

Element	Class	Restrictions and Comments
ProbeEnvelope		
TraceInformation	M*	The first MTA in the domain to receive the probe adds the TraceInformation. When a probe is generated, the originating MTA adds the TraceInformation.
InternalTraceInfo	M/P	This field is mandatory for envelopes transferred between MTAs within a PRMD, and prohibited in messages transferred outside the domain.

13.7.5.3 Reliable Transfer Server (RTS)

In the pUserData of PConnect, the value of applicationProtocol should be 1. This value was chosen because the agreements on intra-domain connections are not strictly P1, nor are they MOTIS. Philosophically, it would be good to choose a new application protocol identifier for these agreements, but this introduces too many practical problems. Since these agreements are closer to P1 than to MOTIS, the value of 1 will be used. This will not cause interworking problems between domains, because the only deviation from P1 is the InternalTraceInfo, which will not be present in messages transferred outside of a domain.

13.8 ERROR HANDLING

This section describes appropriate actions to be taken upon receipt of protocol elements which are not supported in this profile, malformed MPDUs, unrecognized O/R Name forms, content errors, errors in reports, and unexpected values for protocol elements.

13.8.1 MPDU Encoding

The MPDU should have a context-specific tag of 0, 1, or 2. If it does not have one of these tags, it is not possible to figure out who originated the message. Therefore, the way this error is reported is a local matter.

13.8.2 Contents

Once delivery to the UA has occurred, it is not possible to report errors in P2 information to the originator. In addition, it seems unreasonable to insist that the MTA that delivers a message ensure that the P2 content of the message is acceptable. As a result, the handling of content errors is a local matter.

13.8.3 Envelope

This section describes the handling of errors in message envelopes. Some of the error conditions described below may be detected in a recipient's O/R Name. This may limit the reporting MTA's ability to generate a nondelivery notification that accurately reflects the erroneous O/R Name in the ReportedRecipientInfo. This handling of this situation is a local matter.

13.8.3.1 Pragmatic Constraint Violations

In all cases of pragmatic constraint violation, a nondelivery report should be generated with a ReasonCode of unableToTransfer, and a DiagnosticCode of pragmaticConstraintViolation.

13.8.3.2 Protocol Violations

a) If all required protocol elements are not present, a nondelivery report with a ReasonCode of unableToTransfer and a DiagnosticCode of protocolViolation should be generated.

b) If a protocol element is expected to be of one type, but is encoded as another, then a nondelivery report with a ReasonCode of unableToTransfer and a DiagnosticCode of invalidParameters should be generated.

Note: It would be desirable for the CCITT to add a DiagnosticCode of protocolViolation to allow a more meaningful description of this problem. A request for this new DiagnosticCode has been submitted.

13.8.3.3 O/R Names

a) The domain that has responsibility for delivering a message should also have the responsibility to send the nondelivery notification if the message cannot be delivered. Therefore, each MTA should only validate the O/R Names of recipients with responsibility flags set to TRUE. In addition, a nondelivery notification can only be sent if the originator's O/R Name is valid.

b) If any element in the O/R Name is unrecognized or if the CountryName, AdministrationDomainName, and one of PrivateDomainName and OrganizationName (and, for ADMDs,

PersonalName and OrganizationalUnit) are not all present, then a nondelivery report should be generated with a ReasonCode of unableToTransfer, and a DiagnosticCode of unrecognizedORName. If the message can be delivered even though the ORName is invalid, delivery is a local matter. Note, however, that if the message is delivered, the invalid ORName might be propagated through the X.400 system (e.g., by forwarding).

c) If the O/R Name has all of the appropriate protocol elements and the message still cannot be delivered to the recipient, the following DiagnosticCodes may appear in the nondelivery report: unrecognizedORName, ambiguousORName, and uaUnavailable.

13.8.3.4 TraceInformation

a) Since non-relaying domains need not do loop suppression, domains with responsibility for delivering the message need not be concerned about the semantics of the TraceInformation, that is, arrival time and converted EncodedInformationTypes can be provided to the UA without inspection by the MTAs of the domain as long as the TraceInformation is properly encoded according to X.409.

b) When a message is accepted for relay, the relaying domain must check that a TraceInformation SEQUENCE has been added by the domain that last handled the message. If the appropriate TraceInformation was not added, this should be treated as a protocolViolation (section 13.8.3.2).

c) In addition, the relaying domain must check that the information was added in the sequence defined by the rules for adding TraceInformation in the CCITT X.400 Implementor's Guide. If the sequence is invalid,then a nondelivery report should be generated with a ReasonCode of unableToTransfer and a diagnosticCode of invalidParameters.

Note: It would be desirable for the CCITT to add a diagnostic code of invalidTraceInformation to allow a more meaningful description of this problem. A request for this new diagnostic code will be submitted.

13.8.3.5 InternalTraceInfo

This section applies only to MTAs which follow the agreements of section 13.7.

a) When a message is accepted for relay from another MTA in the domain, the relaying MTA must check that an InternalTraceInfo SEQUENCE has been added by the MTA that last handled the message. If the appropriate InternalTraceInfo was not added,

this should be treated as a protocolViolation (section 13.8.3.2).

b) In addition, the relaying MTA must check that the information was added in the sequence defined by the rules for adding TraceInformation in the CCITT X.400 Implementor's Guide. If the sequence is invalid, then a nondelivery report should be generated with a ReasonCode of unableToTransfer and a diagnosticCode of invalidParameters.

Note: It would be desirable for the CCITT to add a diagnostic code of invalidTraceInformation to allow for a more meaningful description of this problem. A request for this new diagnostic code will be submitted.

13.8.3.6 Unsupported X.400 Protocol Elements

The protocol elements defined in X.400 but unsupported by this profile are: the deferredDelivery and PerDomainBilateralInfo parameters of the UMPDUEnvelope, the ExplicitConversion parameter of RecipientInfo, and the alternateRecipientAllowed and contentReturnRequest bits of the PerMessageFlag. Appropriate actions are described below for domains that do not support the protocol elements.

13.8.3.6.1 deferredDelivery

The delivering domain shall do one of the following:

- o deliver at once,
- o hold for deferred delivery,
- o return a nondelivery notification with a ReasonCode of unableToTransfer and a DiagnosticCode of noBilateralAgreement.

13.8.3.6.2 PerDomainBilateralInfo

If a delivering domain receives this element, the element can be ignored.

13.8.3.6.3 ExplicitConversion

If ExplicitConversion is requested the message should be delivered if possible. That is, if the UA is registered to accept the EncodedInformationTypes of the message, then the message should be delivered even though the requested conversion could not be performed along the route. If delivery is not possible, then a nondelivery report should be generated with a ReasonCode of conversionNotPerformed with no DiagnosticCode.

13.8.3.6.4 alternateRecipientAllowed

If a delivering domain receives this element the element can be ignored.

13.8.3.6.5 contentReturnRequest

If a delivering domain receives this element, the element can be ignored.

13.8.3.7 Unexpected Values for INTEGER Protocol Elements

There are three INTEGERs in the P1 Envelope. Appropriate actions are described below for domains receiving unexpected values for Priority, ExplicitConversion, and ContentType.

13.8.3.7.1 Priority

Additional values for Priority have been suggested by at least one group of implementors as upward compatible changes to the X.400 Recommendations. Therefore, if a PRMD receives an unexpected value for Priority, and this value is greater than one byte in length, a nondelivery report should be generated with a ReasonCode of unableToTransfer and DiagnosticCode of invalidParameters. If the value is less than or equal to one byte, the PRMD can either generate a nondelivery report as previously specified or interpret the Priority as normal and deliver or relay the message.

13.8.3.7.2 ExplicitConversion

When an unexpected value is received for ExplicitConversion, it should be handled as in section 13.8.3.6.3.

13.8.3.7.3 ContentType

If the ContentType is not supported by the delivering MTA, then a nondelivery report should be generated with a ReasonCode of unableToTransfer, and a DiagnosticCode of contentTypeNotSupported.

13.8.3.8 Additional Elements

In the absence of multilateral agreements to the contrary, receipt of privately tagged elements and protocol elements in addition to those defined in X.400 will result in a nondelivery report with a ReasonCode of unableToTransfer and a DiagnosticCode of invalidParameters.

13.8.4 Reports

There is no mechanism for returning a delivery or status report due to errors in the report itself. Therefore the handling of errors in reports is a local matter.

13.9 MHS USE OF DIRECTORY SERVICES

13.9.1 Directory Service Elements

a) Recommendation X.400 recognizes the need of MHS users for a number of directory service elements. Directory service elements are intended to assist users and their UAs in obtaining information to be used in submitting messages for delivery by the MTS. The MTS may also use directory service elements to obtain information to be used in routing messages. Some functional requirements of directories have been identified and are listed below.

   - o Verify the existence of an O/R name.
   - o Return the O/R address that corresponds to the O/R name presented.
   - o Determine whether the O/R name presented denotes a user or a distribution list.
   - o Return a list of the members of a distribution list.
   - o When given a partial name, return a list of O/R name possibilities.
   - o Allow users to scan directory entries.
   - o Allow users to scan directory entries selectively.
   - o Return the capabilities of the entity referred to by the O/R name.
   - o Provide maintenance functions to keep the directory up-to-date.

b) In addition to functionality, a number of operational aspects must be considered. These include user-friendliness, flexibility, availability, expandability, and reliability.

c) Currently, these aspects of directory service elements and procedures are under study by both the CCITT and the ISO. Both organizations are committed to the development of a single Directory Service specification for use by MHS and all other OSI based applications.

Given the incomplete nature of the ongoing activities within the CCITT and the ISO, no implementation details will be provided now for MHS use of Directory Services. Implementation agreements for MHS Use of Directory Services will be issued when current activities within the CCITT and the ISO are stable.

13.9.2 Use of Names and Addresses

a) It is recognized that these agreements enable a wide variety of naming and addressing attributes (see section 13.5.3.5 ORName Protocol Elements) wherein each PRMD may adopt particular routing schemes within its domain.

b) With the exception of the intra-domain connection agreements:

These agreements make no attempt to recommend a standard practice for electronic mail addressing.

c) Inter-PRMD addressing may be secured according to practices outside the scope of these agreements, such as:

- o manual directories
- o on-line directories
- o ORName address specifications
- o ORName address translation.

d) Further, each PRMD may adopt naming and addressing schemes wherein the user view may take a form entirely different from the attributes reflected in table 13-12. And, each PRMD may have one user view for the originator form and another for the recipient form, and perhaps other forms of user addressing. In some cases (e.g., receipt notification) these user forms must be preserved within the constraints of these implementation agreements. However, mapping between one PRMD user form to another PRMD user form, via the X.400 ORName attributes of these agreements, is outside the scope of these agreements.

13.10 CONFORMANCE

13.10.1 Introduction

In order to ensure that products conform to these implementation agreements, it is necessary to define the types and degrees of conformance testing that products must pass before they may be classified as conformant. This section defines the conformance requirements and provides guidelines for the interpretation of the results from this type of testing.

This section is incomplete and will be enhanced in future versions of this Agreement. Later versions will reflect the problems of conformance testing and will outline specific practices and

recommendations to aid the development of conformance tests and procedures.

13.10.2 Definition of Conformance

For this section, the term conformance is defined by the following:

a) The tests indicated for this section are intended to establish a high degree of confidence in a statement that the implementation under test (IUT) conforms (or does not conform) to the agreements of this section.

b) Conformance to a service element means that the information associated with the service element is made accessible to the user (person or process) whenever this agreement says that this information should be available.

   Accessible means that information must be provided describing how a user (person or process):

   - causes appropriate information to be displayed, or
   - causes appropriate information to be obtained.

c) Conformance to P1, P2, and RTS as part of an X.400 OSI application requires that only the external behavior of that OSI system adheres to the relevant protocol standards.

   In order to achieve conformance to this section, it is not required that the inter-layer interfaces be available for testing purposes.

d) Conformance to the protocols requires:

   - that MPDUs correspond to instances of syntactically correct data units,
   - MPDUs in which the data present in the fields and the presence (or absence) of those fields is valid in type and semantics as defined in X.400, as qualified by this profile,
   - correct sequences of protocol data units in responses (resulting from protocol procedures).

e) Statements regarding the conformance of any one implementation to this profile are not complete unless a Protocol Implementation Conformance Statement (PICS) is supplied.

f) The term "Implementation Under Test" (IUT) is interchangeable with the term "system" in the definition of conformance, and may refer to:

- o a domain, which may be one or more MTA's with co-located or remote UA's,
- o a single instance of an MTA and co-located UA with X.400 (P1, P2, RTS and session) software,
- o a relaying product with P1, RTS and session software,
- o a gateway product.

g) Claiming Implementation Conformance

- o An implementation which claims to be conformant as an ADMD must adhere to the agreements in sections 13.5 and 13.6.
- o An implementation which claims to be conformant as a PRMD must adhere to the agreements in section 13.5.
- o An implementation which claims to be conformant as a relaying PRMD must adhere to the agreements in section 13.5 and the appropriate sections of 13.7.
- o An implementation which claims to be conformant to the intra-domain connection agreements must adhere to the agreements in section 13.5 and the appropriate sections of 13.7.

13.10.3 Conformance Requirements

13.10.3.1 Introduction

Conformance to this specification requires that all the services listed as supported in sections 13.5, 13.6, and if appropriate, 13.7 of these agreements are supported in the manner defined, in either the CCITT X.400 Recommendations or these agreements.

It is the intention to adopt, where and when appropriate the testing methodology and/or the abstract test scenarios currently being defined by the CCITT X.400 Conformance Group. However, it is recognized that formal CCITT Recommendations relating to X.400 Conformance Testing will not be available until 1988. It is also recognized that aspects of these agreements are outside the scope of the CCITT, and that other organizations will have to provide conformance tests in these cases.

13.10.3.2 Initial Conformance

This section is intended to provide guidelines to vendors who envisage having X.400 products available prior to any formal mechanism, or "Conformance Test Center" being made accessible that would allow for conformance to this product specification to be tested.

It is feasible that vendors and carriers will want to enter bilateral test agreements that will allow for initial trials to be carried out for the purposes of testing initial interworking capabilities. It is equally feasible that for the purposes of testing interoperability, only a subset of this specification will initially be tested.

NOTE: By claiming conformance to this subset of information the vendor or carrier CANNOT claim conformance to this entire specification.

There are two aspects to the requirements, interworking and service, as described in the following sections.

13.10.3.2.1 Interworking

The interworking requirements for conformance implies that tests be done to check for the syntax and semantics of protocol data elements for a system as defined by the classification scheme of sections 13.5.2.1.1 and 13.7.5.2. For a relay system, the correct protocol elements should be relayed as appropriate. For a recipient system, a message with correct protocol elements must not be rejected where appropriate.

13.10.3.2.2 Service

For information available to the recipients via the IPMessage Heading and Body, the following should be made accessible:

- IPMessage ID - only the PrintableString portion of the IPMessageId needs to be accessible.
- subject,
- primaryRecipients,
- copyRecipients,
- blindcopyRecipients,
- authorizingUsers,
- originator,
- inReplyTo,
- replyToUsers,
- importance,
- sensitivity,
- IA5Text Bodypart.

APPENDIX 13A: INTERPRETATION OF X.400 SERVICE ELEMENTS

The work on service element definitions is limited to those that are defined as 'supported' in section 13.5 of this specification. Furthermore it is not the intent of this section to define how information should be made available or presented to a MHS user, nor is it intended to define how individual vendors should design their products. In addition, statements on conformance to a specific service element and the allocation of error codes that are generated as a result of violations of the service should be defined in the sections on conformance and errors as part of the main product specification. The main objective is to provide clarification, where required, on the functions of a service element, and in particular what the original intent of the Recommendations were.

SERVICE ELEMENTS

The following Service Elements defined in X.400 have been examined and require further text to be added to their definitions to represent the proposed implementation of these service elements by the X.400 SIG.

The service element clarifications are to be taken in the context of this profile.

Service elements not referenced in this section are as defined in X.400.

PROBE

A PRMD need not generate probes.

If a probe is addressed to and received by a PRMD, the PRMD must respond with a Delivery Report as appropriate at the time the probe was processed.

DEFERRED DELIVERY

In the absence of bilateral agreements to the contrary, Deferred Delivery and Deferred Delivery Cancellation are local matters (i.e., confined to the originating domain) and need not be provided.

The extension of Deferred Delivery beyond the boundaries of the initiating domain is via bilateral agreement as specified in Section 3.4.2.1 of X.411.

Content Type Indication

It is required that both an originating and recipient domain be able to support P2 content type. The ability for domains to be able to exchange content types other than P2 will depend on the existence of bilateral or multi-lateral agreements.

Original Encoded Information Types Indication

It is required that both an originating and recipient domain be able to support IA5 text. Support for other encoded information types, for the purposes of message transfer between domains, will depend on the existence of bilateral or multi-lateral agreements.

The use of the 'unspecified' form of encoded information type should only be used when the UMPDU content represents an SR-UAPDU or contains an auto-forwarded IM-UAPDU.

The original encoded information type of a message is not meaningful unless a message is converted en route to the recipient. These agreements support only IA5 text, which should not undergo conversion. The original encoded information types should be made accessible to the recipient for upward compatibility with the use of non-IA5 text message body parts.

Registered Encoded Information Types

A UMPDU with an 'unspecified' value for Original Encoded Information Type shall be delivered to the UA.

Delivery Notification

The UAContentID may be used by the recipient of the delivery notification for correlation purposes.

Disclosure of Other Recipients

This service is not made available by originating MTAE's to UAE's, but must be supported by relaying and recipient MTAE's.

By supporting the disclosure of other recipients the message recipient can be informed of the O/R names of the other recipient(s) of the message, as defined in the P1 envelope, in addition to the O/R Descriptors within the P2 header.

These agreements do not support initiation of disclosure of other recipients, but the information associated with it should be made accessible to the recipient for upward compatibility with support for the initiation of this service element.

Typed Body

As defined in X.400 with the addition of the Private Body Types that are to be supported. At present there is no mechanism provided within X.420 that would allow you to respond to reception of an unsupported body type.

Action taken in this situation is a local matter.

Blind Copy Recipient Indication

It should be considered that the recipient's UA acts on behalf of the recipient, and therefore may choose to disclose all BCC recipients to each other. Therefore it is the responsibility of the originating domain to submit two or more messages, depending on whether or not each BCC should be disclosed to each other BCC.

Auto Forwarded Indication

A UA may choose not to forward a message that was previously auto-forwarded. In addition there is no requirement for an IPM UA that does not support non-receipt or receipt notification to respond with a non-receipt notification when a message is auto-forwarded.

Primary and Copy Recipients Indication

It is required that at least one primary recipient be specified; however, for a forwarded message this need not be present. The recipient UA should be prepared to accept no primary and copy recipients to enable future interworking with Teletex, Fax, etc.

Sensitivity Indication

A message originator should make no assumptions as to the semantic interpretation by the recipients UA regarding classifications of sensitivity. For example, a personal message may be printed on a shared printer.

Reply Request Indication

In requesting this service an originator may additionally supply a date by which the reply should be sent and a list of the intended recipients of the reply. If no such list is provided then the initiator of the reply sends the reply to the originator of the message and any recipients the reply initiator wishes to include. The replytoUsers and the replyBy date may be specified without any explicit reply being requested. This may be interpreted by the recipient as an implicit reply request. Note that for an auto-forwarded message an explicit or implicit reply request may not be meaningful.

Body Part Encryption

The original encoded information type indication includes the encoded information type(s) of message body parts prior to encryption by the originating domain. The ability for the recipient domain to decode an encrypted body part is a local matter. Successful use of this facility can only be guaranteed if there exists bilateral agreements to support the exchange of encrypted body parts.

Forwarded IP message Indication

The following use of the original encoded information type in the context of forwarded messages is clarified:

- If forwarding a private message body part the originator of the forwarded message shall set the original encoded information types in the P1 envelope to undefined for that body part.

- The encoded information types of the message being forwarded should be reflected in the new original encoded information types being generated.

- See Appendix 13B on recommended practices for the use of the delivery information as part of Forwarded IP-message.

Multipart Body

It is the intent of multipart bodies to allow for the useful and meaningful structuring of a message that is constructed using differing body part types. For example, it is not recommended that a message made up of only IA5 text should be represented as a number of IA5 body parts, each one representing a paragraph of text.

APPENDIX 13B: RECOMMENDED X.400 PRACTICES

13B.1 RECOMMENDED PRACTICES IN P2

1. ORDescriptor

   Vendors following the NBS/OSI Workshop guidelines shall, whenever possible, generate the ORName portion of an ORDescriptor in ALL IPM heading fields.

2. ForwardedIPMessage BodyParts

   ForwardedIPMessage BodyParts should be nested no deeper than eight. There is no restriction on the number of ForwardedIPMessage BodyParts at any given depth.

3. DeliveryInformation

   It is strongly recommended that DeliveryInformation be supplied in both forwarded and autoforwarded message body parts. DeliveryInformation is useful when a message has multiple forwarded message body parts because without it, the EncodedInformationType(s) of the component forwarded messages cannot be deduced easily. DeliveryInformation is useful for autoforwarded messages because the EncodedInformationType of an autoforwarded message is "unspecified" and the EncodedInformationType(s) of the message cannot be determined easily without it. Absence of the EncodedInformationType(s) makes it difficult for a UA to easily determine whether the message can be rendered.

13B.2 RECOMMENDED PRACTICES IN RTS

1. In the case where S-U-ABORT indicates a temporaryProblem, reestablishment of the session should not be attempted for a "sensible" time period (typically not less than five minutes).

   In instances where this delay is not required or necessary, report a localSystemProblem.

2. S-U-EXCEPTION-REPORT reason codes can be interpreted as follows:

- o receiving ability jeopardized (value 1)
  Possible meaning: The receiving RTS knows of an impending system shutdown.

- o local ss-User error (value 5)
  Possible meaning: <for further study>.

- o irrecoverable procedure error (value 6)
  Possible meaning: the current activity is NOT recoverable.

- o non specific error (value 0)
  Possible meaning: <for further study>.

- o sequence error (value 3): The S-ACTIVITY-RESUME request specified a minor synchronization point serial number which does not match the checkpoint data.

3. For purposes of identifying an MTA during an RTS Open, OSI addressing information should be used. This addressing information is conveyed by lower layer protocols and is reflected by the calling and called SSAP parameters of the S-CONNECT primitives.

MTA validation and identification are related, but separate, functions. The mTAName and password protocol elements of the RTS user data should be used for validation, rather that identification, of an MTA. The RTS initiator and responder may independently require each other to supply mTAName and password.

The CallingSSUserReference parameter of the S-CONNECT primitives should only have meaning to the entity that encoded it and should not be used to identify an MTA.

13B.3 RECOMMENDED PRACTICES WITH X.409

The following practices are recommended for use with X.409.

1. The maximum length of a primitive data element is 256.

2. Bit Strings should be built using primitive form. The constructor form should not be used except in the case of very long Bit Strings (e.g., G3Fax or Voice).

3. All defined bits of a Bit String should be present.

   - Note that, in accordance with X.409, defined bits need not be present; missing bits are assumed to be zero.

   - To ensure upward compatibility, Bit Strings of excess length must also be allowed; the excess bits are ignored.

4. The maximum definite length should be (2**32)-1. <For further study>.

5. It is intended that implementations support upwardly compatible changes to X.409, as defined in Version 3 of the X.400-Series Implementor's Guide, but no guarantees will be made about initial implementations.

6. The concrete encoding of ANY must be a valid X.409 type, and can only be omitted if it is an OPTIONAL element in a SET or SEQUENCE.

13B.4 RECOMMENDED PRACTICES FOR ORName

Table 13.12 stipulates that the StandardAttributeList must contain either PrivateDomainName or OrganizationName. It is recommended that, for both originator and recipients in a private domain, the PrivateDomainName field be used.

It is recommended that there should be a DomainDefinedAttribute to be used in addressing UAs in existing mail systems, in order to curtail the proliferation of different types of DomainDefinedAttributes used for the same purpose. The syntax of this DomainDefinedAttribute conforms to the CCITT Pragmatic Constraints, and thus has a maximum value length of 128 octets and a type length of 8 octets, each of type Printable String. Only one occurrence is allowed.

This DomainDefinedAttribute has the type name "ID" (in uppercase). It contains the unique identifier of the UA used in addressing within the domain. This DomainDefinedAttribute is to be exclusively used for routing within the destination domain (i.e., once routed to that domain via the mandatory components of the StandardAttributeList); any other components of the StandardAttributeList may be provided. If they conflict delivery is not made.

The contents of this parameter need not be validated in the originating domain or any relaying domain, but simply transferred intact to the next MTA or domain.

Class 2 and class 3 MTAs in a PRMD should allow administrators to decide the number of OrganizationalUnits that should appear in user names, instead of

Class 2 and class 3 MTAs in a PRMD should allow administrators to decide the number of OrganizationalUnits that should appear in user names, instead of imposing a software controlled limit which is less than four. This is desirable because when two different vendors impose different limits on the number of OrganizationalUnits in a name, it becomes difficult for the administrator to choose a sensible naming scheme.

There are existing mail systems that include a small set of non-Printable String characters in their identifiers. For these systems to communicate with X.400 messaging systems, either for pass-through service or delivery to X.400 users, gateways will be employed to encode these special characters into a sequence of Printable String characters. This conversion should be performed by the gateway according to a common scheme and before insertion in the ID DDA, which is intended to carry electronic mail identifiers. X.400 User Agents may also wish to perform such conversions.

It is recommended that the following symmetrical encoding and decoding algorithm for non-Printable String characters be employed by gateways. The encoding algorithm maps an ID from an ASCII representation to a PrintableString representation. Any non-printable string characters not specified in the table are covered by the category "other" in the table below.

The principal conversion table for the mapping is as follows:

Tbl. 13B.1 Printable string to ASCII mapping

ASCII Character	Printable String Character
% (percent)	(p)
@ (at sign)	(a)
! (exclamation)	(b)
" (quote mark)	(d)
_ (underline)	(u)
((left paren.)	(l)
) (right paren.)	(r)
other	(3DIGIT)

where 3DIGIT has the range 000 to 377 and is interpreted as the octal encoding of an ASCII character.

To encode an ASCII representation to a PrintableString, the table and the following algorithm should be used:

```
IF current character is in the encoding set THEN
   encode the character according to the table above
ELSE
   write the current character;
continue reading;
```

To decode a PrintableString representation to an ASCII representation, the table and the following algorithm should be used:

```
IF current character is not "(" THEN
   write character
ELSE
   {
   look ahead appropriate characters;
   IF composite characters are in the above table THEN
      decode per above table
   ELSE
      write current character;
   }
continue reading;
```

Class 2 and class 3 MTAs in a PRMD should allow administrators to decide the number of OrganizationalUnits that should appear in user names, instead of imposing a software controlled limit which is less than four. This is desirable because when two different vendors impose different limits on the number of OrganizationalUnits in a name, it becomes difficult for the administrator to choose a sensible naming scheme.

13B.5 POSTAL ADDRESSING

For domains wishing to support postal (or physical) delivery options, the following interim set of "nationally-defined" domain defined attributes are recommended. The CCITT will define Standard Attributes in support of physical delivery in its 1988 Recommendations; this is only an interim solution.

CCITT will also be addressing the services associated with physical delivery. This interim solution does not address the end-to-end service aspects of physical delivery; in particular, the following IPM service elements do not currently extend outside of the X.400 environment:

- alternate Recipient Assignment
- PROBE
- Receipt Notification / Non-Receipt Notifications
- Grade of delivery

"Delivery" means passing a message from the MTS to the physical delivery system (PDS), and not to the user (or user agent).

The following three DDAs are recommended to be used to specify a postal (or physical) address:

CNTRPC - encodes the country and postal code for postal delivery. The DDA value is of the form "Country?Postalcode" (for example, "USA?22096"). The country field is optional, the postal code is optional; the separator ("?") is not. If both country and postal code are missing, this DDA should not be specified.

The country and postal code fields are free-form text.

PDA1 -
PDA 2 - These two DDA (signifying Postal Delivery Address strings 1 and 2) form a 256 character free-form postal address. Fields are separated by a question mark ("?"). There is no implied separator between PDA1 and PDA2. The meaning of the fields are defined by each domain supporting the physical delivery interface. PDA1 contains the first 128 characters, PDA2 the next 128 characters. If the PDA string is less than 128 characters, PDA2 is not used.

For example, if the domain interprets the PDA fields as lines, the address

Mr. John Smith
Conway Steel
123 Main Street
Reston VA 22096

would be encoded as follows:

type = "PDA1" value = "Mr. John Smith?Conway Steel?123 Main Street?Reston VA"
CNTRPC = "?22096"

APPENDIX 13C: RENDITION OF IA5Text AND T61String CHARACTERS

13C.1 GENERATING AND IMAGING IA5Text

The characters that may be used in an IA5String are the graphic characters (including Space), control characters and Delete of the IA5 character repertoire ISO 646.

The graphic characters that may be used with a guaranteed rendition are those related with positions 2/0 to 2/2, 2/5 to 3/15, 4/1 to 5/10, 5/15 and 6/1 to 7/10 in the basic 7-bit code table.

The other graphic characters may be used but have no guaranteed rendition.

The control characters that may be used but have no guaranteed effect are a subset consisting of the format effectors 0/10 (LF), 0/12 (FF) and 0/13 (CR) provided they are used in one of the following combinations:

CR LF	to start a new line
CR FF	to start a new page (and line)
LF .. LF	to show empty lines (always after one of the preceding combinations).

The other control characters or the above control characters in different combinations may be used but have no guaranteed effect.

The character Delete may occur but has no guaranteed effect. The IA5String in a P2 IA5Text BodyPart represents a series of lines which may be divided into pages. Each line should contain from 0 to 80 graphic characters for guaranteed rendition. Longer lines may be arbitrarily broken for rendition. Note that X.408 states that for conversion from IA5Text to Teletex, the maximum line length is 77 characters.

13C.2 GENERATING AND IMAGING T61String

For further study.

APPENDIX 13D: DIFFERENCES IN INTERPRETATION DISCOVERED THROUGH TESTING OF THE MHS FOR THE CeBit 87 DEMONSTRATION

Several interworking problems were discovered through multi-vendor testing. These problems, and recommendations for solutions to them are discussed in this appendix.

13D.1 ENCODING OF RTS USER DATA

The password is defined as an ANY in the X.400 Recommendations, and implementor's groups have decided to use an IA5String for this field. There was some confusion about what the X.409 encoding for this IA5String would be, and the correct encoding is:

```
class:      context specific
form:       constructor
id code:    1
length:     length of contents
contents:   (primitive encoding)
     IA5String:     16
     length:        length of contents
     contents:      the password string

contents: (constructor encoding left as an exercise for the reader)
```

Implementations should be prepared to receive any X.409 type for the password because of its definition as an ANY.

13D.2 EXTRA SESSION FUNCTIONAL UNITS

One vendor proposed more than the required set of functional units on opening the session connection, and the receiver rejected the connection. All debate aside about whether the initiator should have proposed units outside of the required set, or whether the receiver should have rejected the connection, the set of functional units can be negotiated in a straightforward way. The following is recommended.

If the initiator proposes using more than the required set of functional units, the responder should specify the set of functional units that it would like to use (which should include the required set) in the open response. The session implementations will automatically use the intersection of the units proposed by both sides.

If the initiator proposes using less than the required set of functional units, the responder should reject the connection. Unfortunately, there is not an appropriate RefuseReason for rejecting the connection, so instead of refusing the connection in the response to the S-CONNECT, the receiver should issue an S-U-ABORT with an AbortReason of protocolError. Note that it is valid to issue an S-U-ABORT instead of responding to the S-CONNECT. A problem report has been submitted to the CCITT requesting the addition of a RefuseReason for this situation.

If the responder proposes using less than the required set of functional units, the session connection is established before the initiator can check for this. If too few functional units have been proposed, the initiator should abort the connection using S-U-ABORT, with an abort reason of protocolError.

13D.3 MIXED CASE IN THE MTA NAME

The MTA name is frequently exchanged over the telephone when two systems are being configured to communicate with one another. In one such telephone exchange, the casing of the MTA name was not specified, the MTA name consisted of both upper and lower case letters, and one of the implementations compared MTA names for equality in a case sensitive manner. Consequently, connections failed until the problem was detected and repaired. It is recommended that the MTA name be compared for equality in a case insensitive manner, and that the password be compared for equality in a case sensitive manner.

13D.4 X.410 ACTIVITY IDENTIFIER

The X.400 Implementor's Guide recommends that the activity identifier be X.409 encoded, but this is only a recommendation and not a requirement. Consequently, receiving systems cannot assume that the activity identifier will be X.409 encoded.

13D.5 ENCODING OF PER RECIPIENT FLAG AND PER MESSAGE FLAG

In the definition of the PerRecipientFlag in X.411, there is a statement that the last three bits are reserved, and should be set to zero. It is unclear whether those bits are unused in the X.409 encoding. Receivers should accept encodings with either zero or three unused bits. A problem report has been submitted to the CCITT asking for clarification.

Though there is not any statement in X.411 about the last four bits of the PerMessageFlag, some vendors have encoded this with zero unused bits, and some have encoded it with four unused bits. The PerMessageFlag should be encoded with at least four unused bits.

13D.6 ENCODING OF EMPTY BITSTRINGS

There are three valid encodings for an empty bitstring: a constructor of length zero, a constructor of indefinite length followed by the end-of-contents terminator, and a primitive of length one with a zero octet as the value.

13D.7 ADDITIONAL OCTETS FOR BITSTRINGS

Nothing in X.409 constrains an implementation from sending two, three, four, or even more octets for a bitstring that fits into one octet, with the undefined bits set to zero. Note that the number of excess octets is bounded by the pragmatic constraints guidelines of the CCITT X.400 Implementor's Guide for all of the bitstrings in P1.

13D.8 APPLICATION PROTOCOL IDENTIFIER

If a value other that 1 is received in the applicationProtocol of the pUserData in the PConnect, NBS implementations will reject the connection. If CEN/CENELEC implementations receive a value other than 8883 for this field, they will reject the connection. This is an unfortunate state of affairs, because if NBS implementations accept the value of 8883 without supporting the MOTIS service elements, they would be misrepresenting themselves. To make matters worse, CEPT uses a value of 1, but relays MOTIS elements, which means that MOTIS elements will be relayed to implementations using a value of 1 to demonstrate that they do not support MOTIS. Work is continuing to try to find a solution that will allow European implementations to interwork with U.S implementations.

13D.9 INITIAL SERIAL NUMBER IN S-CONNECT

Note: Section 13D.9 original text was removed because it was technically incorrect. Correct text will be supplied in March 1987.

13D.10 CONNECTION DATA ON RTS RECOVERY

It is clarified that the ConnectionData is identical in both the S-CONNECT.request and the S-CONNECT.response. The value of the ConnectionData is the old Session Connection Identifier.

13D.11 ACTIVITY RESUME

If an activity is being resumed on a new session connection, it is not clear from X.410 and X.225 whether all four of the called-ss-user reference, the calling-ss-user reference, the common reference, and the additional reference information should be specified in the S-ACTIVITY-RESUME, or whether one of the ss-user-references should be absent. It is also unclear whether the called-ss-user reference should be identical to the calling-ss-user reference if both are present. Consequently, receivers should be tolerant of this situation. Appropriate problem reports will be submitted to the CCITT asking for clarification.

13D.12 OLD ACTIVITY IDENTIFIER

The Old Activity Identifier in S-ACTIVITY-RESUME refers to the original activity identifier.

13D.13 NEGOTIATION DOWN TO TRANSPORT CLASS 0

For European implementations, X.410 specifies that class 0 transport must be supported. However, it is permissible for an initiator to propose a higher class as the preferred class, provided that class 0 appears as the alternate class in the T-Connect PDU. A responding implementation can choose to use either the preferred or alternate class, but again, must be able to use class 0. In other words, for private to private connections in Europe, class 0 transport is required.

This conflicts with the NBS agreements, since class 0 is only required if one of the partners in a connection is an ADMD.

APPENDIX 13E: WORLDWIDE X.400 CONFORMANCE PROFILE MATRIX

Y CONFORMANCE (E)
implies a conformance problem for European products in the U.S.

Y CONFORMANCE (US)
implies a conformance problem for U.S. products in Europe.

Tbl. 13E.1 Protocol element comparison of RTS

RTS element	NBS	A/311	A/3211	PROBLEM Y/N
PConnect	M	M	M	N
DataTransferSyntax	M 0	M 0	M 0	N
PUserData	M	M	M	N
checkpointSize	H	H	H	N
windowSize	H	H	H	N
dialogueMode	H	H	H	N
connectdata	M	M	M	N
applicationProtocol	R 1	H 1	R 8883	Y A/3211 cannot interwork
ConnectionData				
Open	G	G	?	? A/3211 undefined
Recover	G	H	?	Y Conformance (E)
Open				
RTSUserData	G	G	G	N
Recover				
SessionConnectionID	G	G	G	N
RTSUserData				
MTAName	G	G	G	N
Password	G	G	G	N
null	G	G	G	N
SessionConnectionID				
CallingUserReference	M	M	M	N
CommonReference	M	M	M	N
AdditionalRefInfo	H	H	H	N
PAccept	G	G	G	N
DataTransferSyntax	M 0	M 0	M 0	N

(Continued on next page.)

Tbl. 13E.1 Protocol element comparison of RTS, continued

RTS element	NBS	A/311	A/3211	PROBLEM (Y/N)
PUserData	M	M	M	N
CheckpointSize	H	H	H	N
WindowSize	H	H	H	N
ConnectionData	M	M	M	N
PRefuse	G	G	G	N
RefuseReason	M	M	M	N
SSUserData (in S-TOKEN-PLEASE)	G	G	G	N
AbortInformation (in S-U-ABORT)	G	G	G	N
AbortReason	H	H	H	N
reflectedParameter	X	X	X	N

Tbl. 13E.2 Protocol element comparison of P1

P1 Protocol	NBS	A/311	A/3211	PROBLEM (Y/N)
ORname				
StandardAttributeList	M	M	M	N
DomainDefAttributeList	X	X	X	N
StandardAttributeList				
CountryName	R	R	R	N
		ISO R	R	N
		X.121 H	H	Y Conformance (E)
		Other X	X	Y Prot Vio
AdministrationDomainName	R	R	G	N
... if PrintableString		R	G	N
... if numericString		H	H	Y Conformance (E)
X.121 Address	X	X/R	X	Y Conformance (US)
Terminal ID	X	X/G	X	Y Conformance (US)
PrivateDomainName	G	G	G	N
OrganizationName	G	G	G	N
UniqueUAidentifier	X	X/G	X	Y Conformance (US)
PersonalName.	G	G	G	N
OrganizationalUnit	G	G	G	N
DomainDefinedAttribute	X	X	X	N
Type	M	M	M	N
Value	M	M	M	N
PersonalName				
Surname	M	M	M	N
GivenName	G	G	G	N
Initials	G	G	G	N
GenerationQualifier	G	X	X	Y Conformance (E)
GlobalDomainIdentifier				
CountryName	M	M	M	N
AdministrationDomainName	M	M	G	Y
PrivateDomainIdentifier	R/H	H	R	N
MPDU				
UserMPDU	G	G	G	N
DeliveryReportMPDU	G	G	G	N
ProbeMPDU	H	H	H	N

(Continued on next page.)

Tbl. 13E.2 Protocol element comparison of P1, continued

P1 Protocol	NBS	A/311	A/3211	PROBLEM (Y/N)
UserMPDU				
UMPDUenvelope	M	M	M	N
UMPDUcontent	M	M	M	N
UMPDUenvelope				
MPDUidentifier	M	M	M	N
originatorORname	M	M	M	N
originalEncodedTypes	G	H	H	Y H=illegal
ContentType	M	M	M	N
UAcontentID	H	H	H	N
Priority	G	G	G	N
PerMessageFlag	G	G	G	N
DeferredDelivery	X	X	X	N
PerDomainBilatInfo	X	X	X	N
RecipientInfo	M	M	M	N
TraceInformation	M	M	M	N
MOTIS-> LatestDelivery		X	X	Y Prot Vio
MOTIS-> InternalTraceInfo	<---prohibited --->			N
UMPDUcontent	M	M	M	N
MPDUidentifier				
GlobalDomainIdent	M	M	M	N
IA5string	M	M	M	N
PerMessageFlag				
DiscloseRecipients	H	G @ MTL	H	Y Conformance
		H at UA	?	Y Conformance
ConversionProhibited	G	G	G	N
AlternatRecipAllowed	H	G @ MTL	H	Y Conformance
		H at UA	?	Y Conformance
ContentReturnRequest	X	X	X	
MOTIS-> redirectionProhibited		X	X	Y
PerDomainBilateralInfo				
CountryName	M	M	M	N
AdminDomainName	M	M	G	Y Prot Vio
MOTIS-> PrivateDomainName		X	G	Y Prot Vio
BilateralInfo	M	M	M	N

(Continued on next page.)

Tbl. 13E.2 Protocol element comparison of P1, continued

P1 Protocol	NBS	A/311	A/3211	PROBLEM (Y/N)
DeliveryReportContent				
original MPDUident	M	M	M	N
intermediate Trace	X/G	X	X	Y Conformance (E)
UAcontentID	G	G	G	N
ReportedRecipientInfo	M	M	M	N
returned	H	H	X	Y Conformance (E)
billing information	X	X	X	N
ReportedRecipientInfo				
recipient ORname	M	M	M	N
extensionsIdentifier	M	M	M	N
PerRecipientFlag	M	M	M	N
LastTraceInformation	M	M	M	N
intendedRecipient	H	H	H	N
SupplementaryInfo	X/H	X	X	Y Conformance (E)
MOTIS-> ReassignmentInfo		X	X	Y Prot Vio
MOTIS-> ReassignmentInfo				
MOTIS-> intendedRecipient		X	M	Y Prot Vio
MOTIS-> reasonForReassignment		X	H	Y Prot Vio
LastTraceInformation				
arrival	M	M	M	N
convertedEncInfoTypes	G	G	H	Y Conformance (E)
Report	M	M	M	N
Report				
DeliveredInfo	G	G	G	N
NonDeliveredInfo	G	G	G	N
DeliveredInfo				
delivery	M	M	M	N
TypeofUA	R/H	X	R	N
NonDeliveredInfo				
ReasonCode	M	M	M	N
DiagnosticCode	H	H	H	N
MOTIS-> UAprofileIdentifier		X	X	Y Dropped?
MOTIS-> UAprofileIdentifier		X	X	Y
MOTIS-> ContentType		X	M	Y Dropped?
MOTIS-> EncodedInfoTypes		X	M	Y Dropped?

(Continued on next page.)

Tbl. 13E.2 Protocol element comparison of P1, continued

P1 Protocol	NBS	A/311	A/3211	PROBLEM (Y/N)
ProbeEnvelope				
probe	M	M	M	N
originator	M	M	M	N
ContentType	M	M	M	N
UAcontentID	H	H	H	N
originalEncInfoTypes	G	H	H	Y Conformance (E)
TraceInformation	M	M	M	N
PerMessageFlag	G	G	G	N
ContentLength	H	H	H	N
PerDomainBilatInfo	X	X	X	N
RecipientInfo	M	M	M	N
MOTIS-> InternalTraceInfo	<---prohibited--->			N
RecipientInfo				
RecipientCRname	M	M	M	N
ExtensionIdentifier	M	M	M	N
PerRecipientFlag	M	M	M	N
ExplicitConversion	X	X	X	N
MOTIS-> OriginatorReqAlternatRecip	?	X	X	Y Prot Vio
MOTIS-> ReassignmentInfo	?	X	X	Y Prot Vio
PerRecipientFlag				
ResponsibilityFlag	M	M	M	N
ReportRequest	M	M	M	N
UserReportRequest	M	M	M	N
TraceInformation				
GlobalDomainIdent	M	M	M	N
DomainSuppliedInfo	M	M	M	N

(Continued on next page.)

Tbl. 13E.2 Protocol element comparison of P1, continued

P1 Protocol	NBS	A/311	A/3211	PROBLEM (Y/N)
DomainSuppliedInfo				
arrival	M	M	M	N
deferred	X	X	X	N
action	M	M	M	N
(0=relayed)	G	G	G	N Note: Re-routing not required.
(1=rerouted)	H	H	H	N
MOTIS-> (2=recipientReassigned)		X	H	N
converted	H	G	H	Y Conformance
previous	H	G	G	Y Conformance (Note: G is inconsistent with action (relayed) being "H".)
ORname				
EncodedInformationTypes				
BitString	M	M	M	N
G3NonBasicParameters	X	X	X	N
TeletexNonBasicParams	X	R	X	Y Conformance (US
PresentationAbilities	X	X	X	N
DeliveryReportMPDU	G	G	G	N
DeliveryReportEnvelop	M	M	M	N
DeliveryReportContent	M	M	M	N
DeliveryReportEnvelope				
report	M	M	M	N
originator ORname	M	M	M	N
TraceInformation	M	M	M	N
InternalTraceInfo	<--- prohibited--->			N

Tbl. 13E.3 Protocol element comparison of P2

P2 Protocol	NBS	A/311	A/3211	PROBLEM (Y/N)
UAPDU				
IM_UAPDU	G	G	G	N
SR_UAPDU	X	X	X	N
IM_UAPDU				
Heading	M	M	M	N
Body	M	M	M	N
Heading				
IPmessageID	M	M	M	N
Originator ORname	R	R	R	N
AuthorizingUsers	H	H	H	N
PrimaryRecipients	G	G	G	N
CopyRecipients	G	G	G	N
BlindCopyRecipients	H	H	H	N
InReplyTo	G	G	G	N
Obsoletes	H	H	H	N
CrossReferences	H	H	H	N
Subject	G	G	G	N
ExpiryDate	H	H	H	N
ReplyBy	H	H	H	N
ReplyToUsers	H	H	H	N
Importance	H	H	H	N
Sensitivity	H	H	H	N
Autoforwarded	H	H	H	N
MOTIS-> CirculationList		X	X	Y
MOTIS-> ObsoletingTime		X	X	Y
IPmessageID				
ORname	H	H	H	N
PrintableString	M	M	M	N
ORdescriptor				
ORname	H	H	H	N
FreeFormName	H	H	H	N
TelephoneNumber	H	H	H	N
Recipient				
ORdescriptor	M	M	M	N
ReportRequest	X	X	X	N
ReplyRequest	H	H	H	N

(Continued on next page.)

Tbl. 13E.3 Protocol element comparison of P2, continued

P2 Protocol	NBS	A/311	A/3211	PROBLEM (Y/N)
MOTIS-> CirculationMember	X	X	X	N
MOTIS-> checkmark	X	X	M	Y Conformance (US)
MOTIS-> membername	X	X	M	Y Conformance (US)
MOTIS-> OBsoletingTime	X	X	X	N
MOTIS-> Time	X	X	H	N
MOTIS-> IP_MessageID	X	X	H	N
Body				
BodyPart	G	M	M	Y Conformance (US)
SR_UAPDU				
NonReceipt	H	H	H	N
Receipt	H	H	H	N
Reported	M	M	M	N
ActualRecipient	R	R	R	N
IntendedRecipient	H	H	H	N
Converted	X	X	X	N
MOTIS-> CirculationStatus	X	X	X	N
NonReceiptInformation				
Reason	M	M	M	N
NonReceiptQualifier	H	H	H	N
=expired (value)	X	X	H	Y Conformance (US)
=obsoleted (value)	X	X	H	Y Conformance (US)
=subscriptionTerminated	X	X	H	Y Conformance (US)
MOTIS-> =timeobsoleted (value)	X	X	X	N
Comments	H	H	H	N
returned	H	X	X	Y Conformance (E)
ReceiptInformation				
Receipt	M	M	M	N
TypeOfReceipt	H	H	H	N
SupplementaryInfo	X	X	X	N

(Continued on next page.)

Tbl. 13E.3 Protocol element comparison of P2, continued

P2 Protocol	NBS	A/311	A/3211	PROBLEM (Y/N)
BODYPART SUPPORT				
o IA5 Text	G	G	G	N
o TLX	X	X	X	N
o Voice	X	X	X	N
o G3FAX	X	X	X	N
o TIFO	X	X	X	N
o TTX	X	X/H	X	Y Conformance (US)
o VideoTex	X	X	X	N
o NationallyDefined	X	X	X	N
o Encrypted	X	X	X	N
o ForwardedIPmessage	H	H	H	N
o SFD	X	X	X	N
o TIFI	X	X	X	N

14. RESERVED FOR NEXT VERSION OF X.400

15. DIRECTORY SERVICES PROTOCOLS

To be completed.

16. ISO VIRTUAL TERMINAL PROTOCOL

16.1 INTRODUCTION

The NBS/OSI Workshop Virtual Terminal (VT) SIG is making implementation agreements for the OSI Basic Class VT Service and Protocol, ISO/DIS 9040 and 9041, including the first addenda to both 9094 and 9041, subject to these addenda reaching a stable state (i.e., DAD) by the implementation agreement capability freeze date.

These implementation agreements fall into the following categories.

- o Functionality to be implemented, i.e., subsets, etc..
- o Identification and specification of VT profiles to be supported by conforming implementations.
- o Agreements with regard to implementation issues not specified in ISO/DIS 9040 and 9041 and their addenda.
- o Resolution of problems with ISO 9040 and 9041 identified during implementation.
- o Statement of requirements to meet conformance to these agreements.

These implementation agreements will be aligned with any changes made in progressing ISO 9040 and 9041 and their addenda from DIS to IS.

16.2 SCOPE AND FIELD OF APPLICATION

To be determined.

16.3 STATUS

The NBS workshops Virtual Terminal Implementation Agreements are being done in two phases. The items below provide the status of each phase.

1) The Phase I Virtual Terminal Agreements are expected to be completed by July 1987. The Phase I agreements will be based on ISO/DIS 9040 and 9041. The first addenda to 9040 and 9041 will form a part of the Phase I agreements, subject to their attaining DAD status by the capability freeze date of July 1987. The Status of ISO 9040 and 9041 will be either IS or 2nd DIS by the capability freeze date.

2) The Phase II agreements will be completed at an unspecified future date, and will be based on IS Virtual Terminal documents.

3) It is intended that Phase II agreements be compatible with Phase I agreements, provided no changes are made to the Standards (in progressing from DIS to IS) to make this goal impossible.

4) The Phase I Agreements assume that the changes to the mapping of Protocol

Elements in ISO/DIS 9041, as suggested by ANSI, will be accepted.

16.4 ERRATA

16.5 SERVICES

16.5.1 Services Provided

16.5.1.1 Basic Class Service Subsets

The VT-A and VT-B service subsets as described in ISO/DIS 9040 have been selected. The VT-C subset will not be used. The following service facilities provided in subsets A and B have been selected.

- o Association Establishment
- o Association Termination
- o Switch Profile Negotiation
- o Data Transfer
- o Delivery Control - Simple Delivery control only has been selected. Quarantine delivery control will not be used.
- o Access Right Management

16.5.1.2 Extended Facility Set

The extended service facility set, as described in the first addendum to ISO/DIS 9040, has been selected.

16.5.1.3 Modes of Operation

Both Asynchronous and Synchronous modes of operation have been selected.

16.5.1.4 Access Rights

All types of Access Right mechanisms, as specified in ISO/DIS 9040 and its addendum, will be supported.

16.5.2 Underlying Services Assumed

The following lower layer services are assumed.

- o ACSE
- o PRESENTATION - Kernel plus Half-Duplex and Symmetric Synchronization Functional Units
- o SESSION - Kernel plus Typed Data plus Half-Duplex, Full Duplex and Symmetric Synchronization Functional Units
- o TRANSPORT - Class 4

16.5.3 Service Profiles

The two Default Virtual Terminal Profiles, one for each mode of operation, as specified in ISO/DIS 9040 will be supported. Additionally, the following profiles have been selected.

- o TELENET
- o PAGE
- o CCITT X.3 PAD Compatible Operation
- o TRANSPARENT
- o FORMS
- o SCROLL

16.6 PROTOCOL

16.6.1 Protocol Elements

All Protocol Elements supported by the VT-B subset have been selected.

16.6.2 Mapping of Protocol Elements

Mapping of protocol elements on to underlying ACSE or Presentation Services is as defined in ISO/DIS 9041.

16.6.3 Protocol Data Unit Structure

Protocol data unit structure is as defined in ISO/DIS 9041.

16.7 CONFORMANCE

To be determined.

16.8 TEST REQUIREMENTS

To be determined.

16.9 TELNET PROFILE

This profile provides support for TELNET-like operation for users of the ISO Virtual Terminal Service. Control object updates are meant to be equivalent to TELNET commands.

Bit 0 of the DISPLAY-SIGNAL and KEYBOARD-SIGNAL control objects corresponds to the TELNET AYT (are you there) command, bit 1 of the DISPLAY and KEYBOARD control objects to the TELNET AO (abort output) command, bit 1 of the SYNC control object to the TELNET AO/SYNC command, bit 2 of the DISPLAY and KEYBOARD control objects to the TELNET IP (interrupt process) command, bit 2 of the SYNC control object to the TELNET IP/SYNC command, and bit 3 of the SYNC control object to the TELNET Break command. The TELNET EC (erase character) command should be mapped to an erase-x-array-forward update and the TELNET EL (erase line) command to an erase-full-x-array update. Users of this profile should refer to the TELNET specification (MIL-STD-1782) for semantics of the TELNET commands.

The profile is defined as follows:

```
Display-objects =
{
        {    display-object-name = DISPLAY,
             object-access-right = WACA,
             dimensions = 2
                       x-dimension =
                       {

                            x-bound = r1,
                            x-forward = 0,
                            x-backward = r1,
                            x-absolute = "not permitted",
                            x-window - r1

                       },
                       y-dimension =

                       {
                            y-bound = "unbounded",
                            y-forward = 1,
                            y-backward = 0,.
                            y-absolute = "not permitted",
                            y-window = 1

             repertoire-assignment - <"7-bit ASCII", "binary">
        },
        {
             display-object-name = KEYBOARD,
             object-access-right = WACI,
             dimensions = 2,
                       x-dimension =
                       {
                            x-bound = "unbounded",
                            x-forward = 0,
                            x-backward = r1,
                            x-absolute = "not permitted",
                            x-window - "not defined"
                       },
                       y-dimension =
                       {
                            y-bound = "unbounded",
                            y-forward = 1,
                            y-backward = 0,
                            y-absolute = "not permitted",
                            y-window = 1
                       },
             repertoire-assignment = <"7-bit ASCII", "binary">
        }
Control-object =
{
        {
```

```
            CO-name = SYNC,
            CO-access = "not-subject-to-access-control",
            CO-category = boolean,
            CO-size = 4
        },

        {
            CO-name = DISPLAY-SIGNAL,
            CO-access = WACA,
            CO-category = boolean,
            CO-size = 4,
            CO-trigger = not selected
        }

        {
            CO-name = KEYBOARD-SIGNAL,
            CO-access = WACI,
            CO-category = boolean,
            CO-size = 4
            CO-trigger = not selected
        }

        {
            CO-name = VT-WACI-ECHO, *( echo control object )*
            CO-access = WACI,
            CO-trigger = selected,
            CO-category = boolean,
            CO-size = 1
        }

        {
            CO-name = VT-WACA-ECHO, *( echo control object )*
            CO-access = WACA,
            CO-trigger = selected,
            CO-category = boolean,
            CO-size = 1
        }
},
Device-objects =
{
        {
            device-name = DISPLAY-DEVICE,
            device-display-object = DISPLAY,
            device-default-CO-initial-value = "true" *( initially "on"  )*
            device-repertoire-assignment = <"7-bit ASCII", "binary">
            device-minimum-x-array-length = 1, *( no constraint )*
            device-minimum-y-array-length = 1, *( no constraint )*
            device-control-object = {SYNC, DISPLAY-SIGNAL},
            device-termination-event-list = NULL,
            *( other device parameters assume default values or
                 are not required )*
        },
        {
```

```
            device-name = KEYBOARD-DEVICE,
            device-display-object = KEYBOARD,
            device-default-CO-access = WACI,
            device-default-CO-initial-value = "true", *( initially "on" )*
            device-minimum-x-array-length = 1, *( no constraint )*
            device-minimum-y-array-length = 1, *( no constraint )*
            device-control-object = {SYNC, KEYBOARD-SIGNAL},
            device-termination-event-list = NULL,
                 *( other device parameters assume default values or
                      are not required )*
        }
}

type-of-delivery-control = simple-delivery-control.
```

NOTES

1. The profile parameter r1 is the size of the x dimension for the display object that is mapped to the terminal screen. It is a mandatory parameter.

2. An attribute update can be used to toggle between the "7 bit ASCII" and "binary" character sets. This is not quite equivalent to the TELNET BINARY option because the new character set is unilaterally asserted for the display object rather than changed by a bilateral negotiation.

3. The echo control objects are used to emulate TELNET's echo negotiation. The two binary control objects can be used to negotiate whether the initiator (terminal) side will echo characters to the local terminal. Updating one control object can be interpreted as a request to start local echo ("true") or stop local echo ("false"). The peer VT-USER should then update the other control object to the same value to accept the requested change or to the opposite value to reject it.

4. Option negotiation in TELNET can take place at any time during a session and modifies option settings one at a time. As subset C of VTP is not supported by the NBS/OSI implementor's agreements, negotiation of TELNET options other than echo and character set is not supported by this profile.

5. The x-forward parameter is 0 for both display objects so that the x address of the pointer can be moved forward only by implicit pointer addressing.

6. Because x-absolute is "not permitted", it is anticipated that upon line termination a pointer relative addressing update rather than a "next x-array" will be used to set the pointer to the beginning of the next line.

7. No termination event list is specified so that data buffering and delivery can be controlled according to context. If local echoing is enabled, the local "newline" sequence should trigger delivery. Otherwise a timeout or buffer length may be used to trigger delivery.

8. Use of this profile is meant to resemble TELNET NVT with the "suppress go-ahead" option negotiated.

This page is blank.

17. OFFICE DOCUMENT ARCHITECTURE AND INTERCHANGE FORMAT

To be completed.

18. PERFORMANCE

To be completed.

19. SECURITY

The Security Architecture specified in ISO 7498/Part 2 - Security Architecture (as presented in ISO/TC 97/SC 21/N1528) shall be used as a basis for further work in the Special Interest Group on Security.

The security services that are to be implemented first shall include confidentiality, integrity, authentication and access control. Non-repudiation of the source shall also be included for consideration for implementation. These services are defined and discussed in more detail in ISO 7498/Part 2 - Security Architecture.

REFERENCES

NBS

FIPS 107, Local Area Networks: Baseband Carrier Sense Multiple Access with Collision Detection Access Method and Physical Layer Specifications and Link Layer Protocol, NTIS, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.

FIPS 100, Interface Between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE) For Operation With Packet-Switched Data Communications Networks, NTIS, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.

ICST/SNA-85-10, Implementation Agreements Among Participants of OSINET, edited by Jerry Mulvenna, National Bureau of Standards.

IEEE

IEEE Project 802, Local Area Network Standards, P802.2 Logical Link Control, November, 1982.

IEEE Project 802, Local Area Network Standards, IEEE Standard 802.4, Token - Passing Bus Access Method and Physical Layer Specification.

IEEE Project 802, Local Area Network Standards, IEEE Standard 802.3, CSMA/CD Access Method and Physical Layer Specification.

Binary Floating Point Arithmetic (ANSI Approved), IEEE 754, March 21, 1985, Institute of Electrical and Electronics Engineers.

The above documents may be obtained from: IEEE Standards Office, 345 East 47th Street, New York, N.Y. 10017.

IEC

Binary Floating Point Arithmetic for Microprocessor Systems, IEC 559, First Edition, International Electrotechnical Commission, June 20,1982.

ISO

Addendum to DIS 8473 Covering Provision of the Connectionless-Mode Subnetwork Service, ISO/TC97/SC 6/N3453.

Network Service Definition, DIS 8348, ISO/TC97/SC6 N2990.

Addendum to the Network Service Definition Covering Connectionless Data Transmission, DIS 8348 DAD1, N3152.

Addendum to the Network Service Definition Covering Network Layer Addressing, DP 8348 DAD2, N3134.

Internal Organization of the Network Layer, WD, N3141.

Protocol for Providing the Connectionless Network Service, DIS 8473, N3154.

Information Processing Systems - Open Systems Interconnection - Transport Service Definition, ISO IS8072, 1984.

Information Processing Systems - Open Systems Interconnection - Transport Protocol Specification, ISO IS8073, 1984.

Information Processing Systems - Open Systems Interconnection - Session Service Definition, ISO DIS8326, 1984.

Information Processing Systems - Open Systems Interconnection - Session Protocol Specification, ISO DIS8327, 1984.

Information Processing Systems - Open Systems Interconnection - File Transfer Access and Management Part 1: General Description, ISO DP8571/1, TC97/SC16 N 1669, February 1984.

Information Processing Systems - Open Systems Interconnection - File Transfer Access and Management Part II: The Virtual Filestore, ISO DP 8571/2, TC97/SC16 N1670, February 1984.

Information Processing Systems - Open Systems Interconnection - File Transfer Access and Management Part III: Service Definition, ISO DP8571/3, TC97/SC16 N1671, February 1984.

Information Processing Systems - Open Systems Interconnection - File Transfer Access and Management Part IV: Protocol Specification, ISO DP8571/4, TC97/SC16 N1672, February 1984.

Data Communication - X.25 Packet Layer Specification for Data Terminal Equipment, ISO/TC 97/SC 6 N 2641, ISO/DP 8208, 1983.

7-bit Coded Character Set for Information Processing Interchange, ISO-646, 1973.

Information Interchange--Representation of Local Time Differentials, ISO-3307, 1975.

Draft Network Layer Management Protocol for the exchange of routing information between end systems and intermediate systems ISO/TC97/SC6/3862 January 1986.

Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management Part I: General Description, ISO DIS8571/1, TC97/SC21 N2371, August 1986.

Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management Part II: The Virtual ISO DIS8571/2, TC97/SC21 N2372, August 1986.

Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management Part III: File Service Definition, ISO DIS8571/3, TC97/SC21 N2373, August 1986.

Information Processing Systems - Open Systems Interconnection - File Transfer, Access and management Part IV: File Protocol Specification, ISO DIS8571/4, TC97/SC21 N2374, August 1986.

Information Processing Systems - Open Systems Interconnection - Connection-Oriented Presentation Service Definition, ISO DIS8822, TC97/SC21 N1594, May 1986.

Information Processing Systems - Open Systems Interconnection - Connection-Oriented Presentation Protocol Specification, ISO DIS8823, TC97/SC21 N1594, May 1986.

Information Processing Systems - Open Systems Interconnection - Service Definition for Common Application Service Elements - Part 2: Association Control, ISO DIS8649/2, TC97/SC21 N1493, May 1986.

Information Processing Systems -- Open Systems Interconnection - Protocol Specification for Common Service Elements Part 2: Association Control, ISO DIS8650/2, TC97/SC21 N1494, May 1986.

Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), DIS 8824, Oct., 1985.

Information Processing Systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1), DIS 8825, Oct., 1985.

The above documents may be obtained from:

Frances E. Schrotter
ANSI
ISO TC97/SC6 Secretariat
1430 Broadway
New York, N.Y. 10018

CCITT

X.25 Interface Between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE) for Terminals Operating in the Packet Mode on Public Data Networks.

X.400 (Red Book, 1984), Message Handling Systems: System Model-Service Elements.

X.401 (Red Book, 1984), Message Handling Systems: Basic Service Elements and Optional User Facilities.

X.408 (Red Book, 1984), Message Handling Systems: Encoded Information Type Conversion Rules.

X.409 (Red Book, 1984), Message Handling Systems: Presentation Transfer Syntax and Notation.

X.410 (Red Book, 1984), Message Handling Systems: Remote Operations and Reliable Transfer Server.

X.411 (Red Book, 1984), Message Handling Systems: Message Transfer Layer.

X.420 (Red Book, 1984), Message Handling Systems: Interpersonal Messaging User Agent Layer.

X.430 (Red Book, 1984), Message Handling Systems: Access Protocol for Teletex Terminals.

X.214 (Red Book, 1984), Transport Service Definition for Open Systems Interconnection for CCITT Applications.

X.224 (Red Book, 1984), Transport Protocol Specification for Open Systems Interconnection for CCITT Applications.

X.215 (Red Book, 1984), Session Service Definition for Open Systems Interconnection for CCITT Applications.

X.225 (Red Book, 1984), Session Protocol Specification for Open Systems Interconnection for CCITT Applications.

X.400 - Series Implementor's Guide (Version 3, 1986).

The above documents may be obtained from: International Telecommunications Union, Place des Nations, CH 1211, Geneve 20 SWITZERLAND.

Miscellaneous

[Edge 84] S. W. Edge, An Adaptive Timeout Algorithm for Retransmission Across a Packet Switching Network, ACM Computer Communications Review, Vol. 14, No. 2, June 1984.

[Jain 85] R. Jain, Divergence of Timeout Algorithms for Packet Retransmission, Proceedings IEEE Computer Communications Conference, Phoenix March 28-29, 1986.

[Mill 83] D. L. Mills, Internet Delay Experiments, DARPA Network Working Group RFC #889, December 1983.

ADDENDUM 1

Note on FTAM and X.400 Character Sets

On July 21, 1986, a group of twelve individuals from the FTAM and X.400 SIGs met to resolve differences in recommended use of character sets. The following was agreed (in favor, 9; opposed, 2; abstaining, 1) by these individuals:

"Both SIGs should implement IA5 for the current phase of development, and independently support expanded character sets. It is recommended that the FTAM and X.400 SIGs support both 8859/1 and 6937/2 in the next phase of their agreements."

Neither SIG brought forward to the plenary on Thursday, July 24, 1986, a recommendation on this issue. However, it was raised for plenary discussion. The plenary felt (in favor, 22; opposed, 3; abstaining, 1) that this information should be carried in some form in this document in addition to inclusion in the minutes. Hence, it is included as an addendum so as to keep the information associated with this document while showing that it has not been accepted for inclusion in the main body of this document.

Index

You will receive the documents from the next workshop by either attending the workshop or completing and returning the form below.

READER RESPONSE FORM

Please retain my name for the next mailing of the NBS/OSI Implementors Workshop.

NAME _____

ADDRESS _____

_____

_____

PHONE NO. _____

Mail this page to: Sara Arneson
National Bureau of Standards
Bldg. 225/A216
Gaithersburg, MD 20899

NBS-114A (REV. 2-8C)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NBSIR 86-3385-4	2. Performing Organ. Report No.	3. Publication Date March 1987

4. TITLE AND SUBTITLE

Implementation Agreements for Open Systems Interconnection Protocols

5. AUTHOR(S)

Robert Rosenthal, Editor

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)

NATIONAL BUREAU OF STANDARDS
U.S. DEPARTMENT OF COMMERCE
GAITHERSBURG, MD 20899

7. Contract/Grant No.

8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

10. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

This document records current agreements on implementation details of Open Systems Interconnection protocols among the organizations participating in the NBS/OSI Workshop Series for Implementors of OSI Protocols. These decisions are documented to facilitate organizations in their understanding of the status of agreements. This is a standing document that is updated after each workshop (about every two and one-half months). A reference list of standards and a list of contributing organizations are included in the Appendix.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

local area networks; NBS/OSI Workshop; network protocols; Open Systems Interconnection; OSINET; testing protocols

13. AVAILABILITY

☐ Unlimited

☒ For Official Distribution. Do Not Release to NTIS

☐ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

☐ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

15. Price